nRF51422 Development Kit

nRF51422

User Guide v1.1

Copyright © 2013 Nordic Semiconductor ASA. All rights reserved.
Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

Contents

1 Introduction 3
1.1 Minimum requirements 3
1.2 External resources 3
1.3 Writing conventions 3
1.4 Development Kit release notes 3

2 Kit content 4
2.1 nRF51422 Development Kit hardware content 4
2.2 Downloadable content 5

3 Quick start 6
3.1 Install the nRF51422 connector module 7
3.2 ANT Link demo 10

4 Development kit configuration 12
4.1 Development environment 12
4.2 Kit setup 15

5 Hardware description 17
5.1 nRFgo nRF51422 DK modules 17
5.2 nRFready ANT USB dongle (nRF2779) 24

6 Flash programming and application development 26
6.1 Programming and erasing flash using nRFgo Studio 26
6.2 Application development 29

7 Debugging the nRF51422 38
7.1 nRF51422 debug features and precautions 39

8 Software Development Kit 42
8.1 Installing the nRF51 SDK 42

9 Troubleshooting 43

Appendix A: Installing drivers and configuring KEIL projects for the SEGGER debugger 45

1 Introduction

The nRF51422 Development Kit (DK) for ANT provides a complete solution for developing, testing, and evaluating the nRF51422 device.

nRF51422 is part of the nRF51 series which offers a range of ultra-low power System on Chip (SoC) solutions for your 2.4 GHz wireless products.

1.1 Minimum requirements

- nRFgo Starter Kit
- nRFgo Studio v1.14 or later
- Computer with a minimum of 2 USB ports
- Windows XP 32 bit (SP3) or Windows 7

1.2 External resources

- Keil MDK-ARM Lite v4.54 or later https://www.keil.com/demo/eval/arm.htm
- J-Link Software v4.52b or later http://www.segger.com/jlink-software.html

1.3 Writing conventions

This User Guide follows a set of typographic rules that makes the document consistent and easy to read. The following writing conventions are used:

- Commands are written in `Lucida Console`.
- Pin names are written in **Consolas.**
- File names and user interface components are written in **bold**.
- Internal cross references are italicized and written in ***semi-bold.***

1.4 Development Kit release notes

Date	Kit version	Description
February 2013	2.0	Fixed known issue in kit v1.0.
November 2012	1.0	**Known issues** PCA10003 v1.0: • The antenna matching network and layout on these boards is suitable for applications using TX output power 0 dBm or less. These boards are not suitable for applications using +4 dBm TX output power.

2 Kit content

The nRF51422 DK consists of hardware and access to software components, documentation, and design files from www.nordicsemi.com. The S210 nRF51422 SoftDevice is pre-programmed on the nRF51422 and cannot be deleted or modified.

2.1 nRF51422 Development Kit hardware content

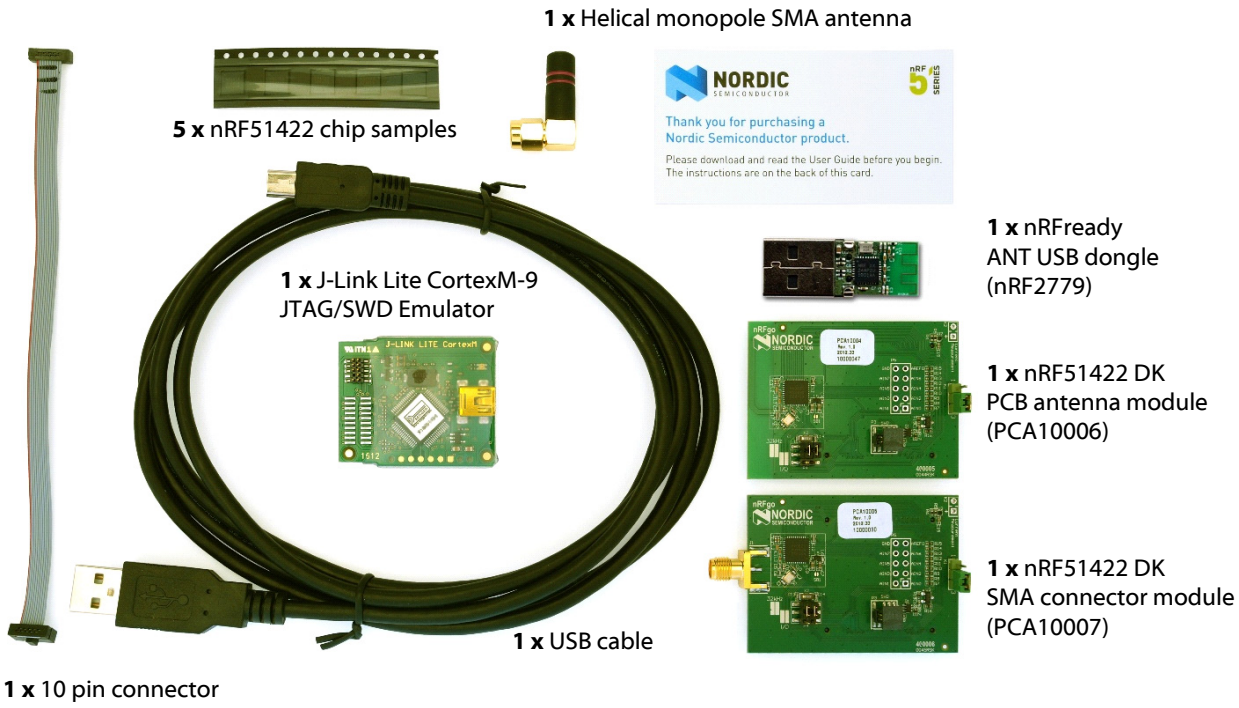

1 x Helical monopole SMA antenna
5 x nRF51422 chip samples
NORDIC SEMICONDUCTOR
Thank you for purchasing a Nordic Semiconductor product.
Please download and read the User Guide before you begin. The instructions are on the back of this card.
1 x J-Link Lite CortexM-9 JTAG/SWD Emulator
1 x nRFready ANT USB dongle (nRF2779)
1 x nRF51422 DK PCB antenna module (PCA10006)
1 x nRF51422 DK SMA connector module (PCA10007)
1 x USB cable
1 x 10 pin connector

Figure 1 *nRF51422 DK hardware content*

2.2 Downloadable content

The nRF51422 Development Kit includes firmware source code, documentation, hardware schematics, and layout files. To access these files, log in to your My Page account, enter your product key, and download the files. Instructions can be found in ***chapter 3 on page 6***.

2.2.1 nRF51422 DK software content

- nRFgo Studio
- nRF51 Software Development Kit (SDK)
  - Precompiled HEX files
  - Source code
  - Keil ARM project files
- ANTware II

2.2.2 nRF51422 DK documentation

- *nRF51422 Development Kit User Guide*
- *nRF51 Series Reference Manual*
- *nRF51422 PS*
- *S210 nRF51422 SoftDevice Specification*
- *nRF51 SDK*
- ANT API documentation
- *nRF51422 PAN*

2.2.3 Schematics, Bill of Materials, PCB layout files, and production files

The ZIP file and its subdirectories contain the hardware design files for the nRF51422 DK.

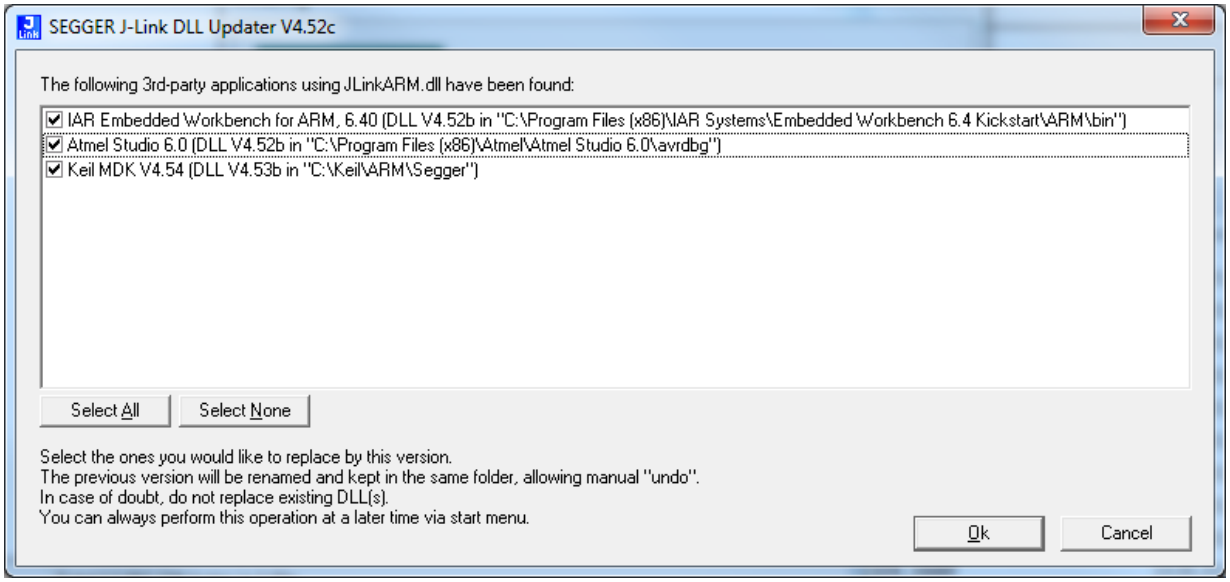
- Altium Designer files
- Schematics
- PCB layout files
- Production files
  - Assembly drawings
  - Drill files
  - Gerber files
  - Pick and Place files
  - Bill of Materials

3 Quick start

This section shows you how to set up the nRF51422 Development Kit and provides example applications to help you start programming your device.

Register, download, and install

1. If you have Keil MDK-ARM Lite already installed, go to step 2. Otherwise, download and install Keil MDK-ARM Lite from https://www.keil.com/demo/eval/arm.htm to your hard drive. Keil downloads to **c:\Keil** unless you change the location when installing.
2. Download and run the J-Link Software and documentation pack for Windows from http://www.segger.com/jlink-software.html. The serial number from your SEGGER J-Link hardware is needed to identify your device and can be found printed on the chip on the J-Link Lite emulator board.
3. During installation you will be prompted to select the IDE that should be updated with the latest SEGGER DLLs. Check the box for **Keil MDK** and any other IDEs you want to use with SEGGER.

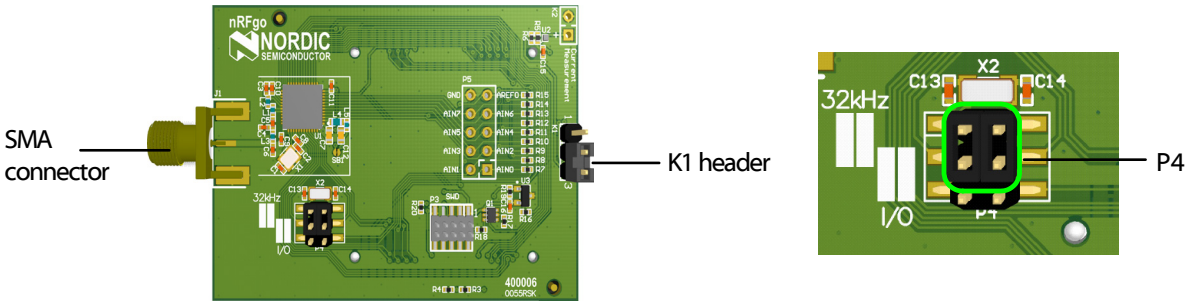

SEGGER J-Link DLL Updater V4.52c
The following 3rd-party applications using JLinkARM.dll have been found:
IAR Embedded Workbench for ARM, 6.40 (DLL V4.52b in "C:\Program Files (x86)\IAR Systems\Embedded Workbench 6.4 Kickstart\ARM\bin")
Atmel Studio 6.0 (DLL V4.52b in "C:\Program Files (x86)\Atmel\Atmel Studio 6.0\avrdbg")
Keil MDK V4.54 (DLL V4.53b in "C:\Keil\ARM\Segger")
Select All
Select None
Select the ones you would like to replace by this version.
The previous version will be renamed and kept in the same folder, allowing manual "undo".
In case of doubt, do not replace existing DLL(s).
You can always perform this operation at a later time via start menu.
Ok
Cancel

4. Go to www.nordicsemi.com and log in to your Nordic My Page account.
5. Select **My Products** from the left menu. This takes you to the My Products page.
6. Enter the product key (included with this kit) into the **Product Key** field and click **Add**.
7. From the **Add product** box, select the product name and click **Add**.
8. Click the **Downloads** link in the **Overview, My Products** table.
9. Download and run the nRF514 SDK installer. Make sure to choose the **Keil MDK-ARM** installer option.

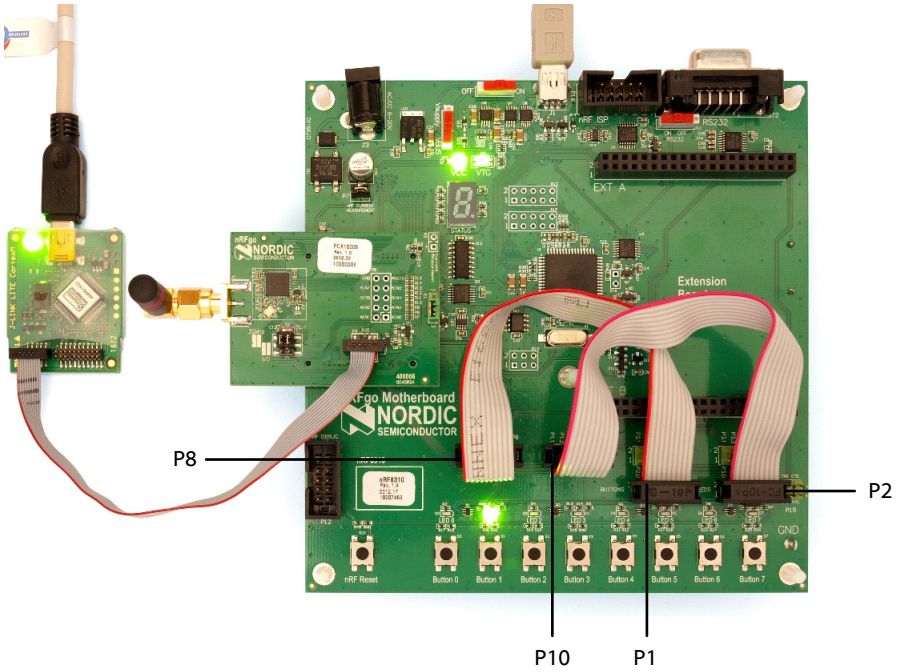
3.1 Install the nRF51422 connector module

Connect the hardware

1. Screw the antenna into the SMA connector on the nRF51422 DK SMA (PCA10007) connector module.
2. Ensure that header **K1** has a jumper connecting pin 2 and 3. The two upper rows of **P4** should have two jumpers placed vertically matching the 32 kHz marking.

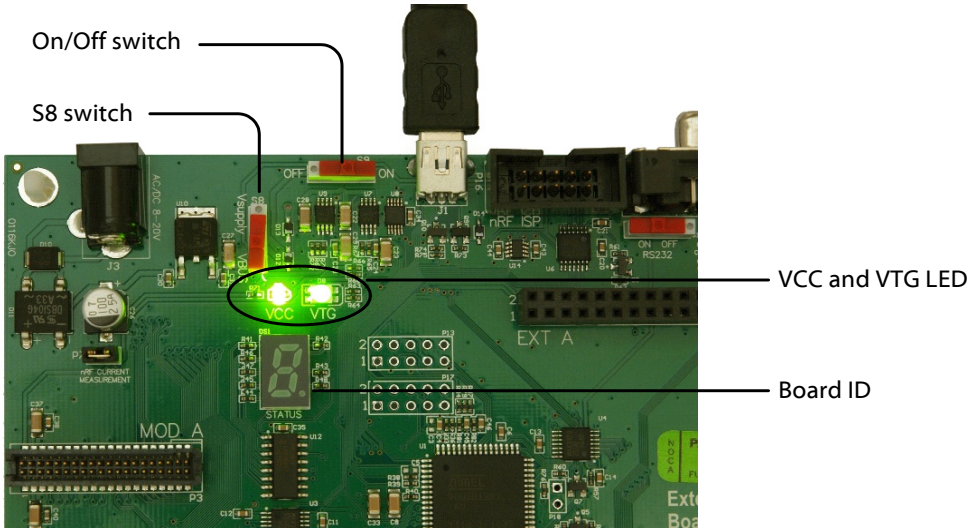


3. Plug the nRF51422 DK SMA connector module into the nRFgo Starter Kit Motherboard (nRF6310).
4. Connect a USB cable from the Motherboard to your computer.
5. Using two 10 pin flat cables (2.54 mm, included in the nRFgo Starter Kit), connect one cable between **P8** (PORT0) and **P1** (BUTTONS) and one between **P10** (PORT1) and **P2** (LEDS) on the nRFGo Motherboard. Make sure the red marking on the cable is always connected to pin 1, as shown in the figure below.
6. Connect the SEGGER J-Link board to the nRF51422 module with the 10 pin flat cable (1.27 mm, provided in the Development Kit).
7. Connect a USB cable from the J-Link board to your computer.

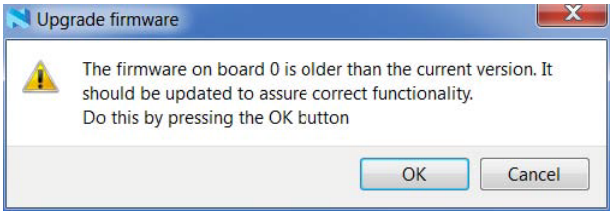


Turn on and set the supply voltage

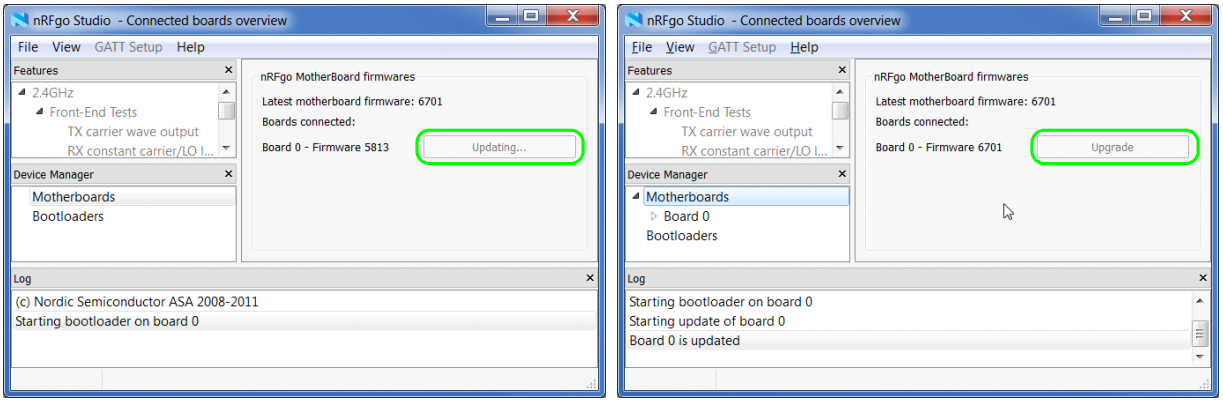
1. Slide the **S8** switch on the Motherboard to **VBUS**.
2. Slide the **On/Off** switch on the Motherboard to **ON**.
3. The **VCC** and **VTG** LEDs will light up.



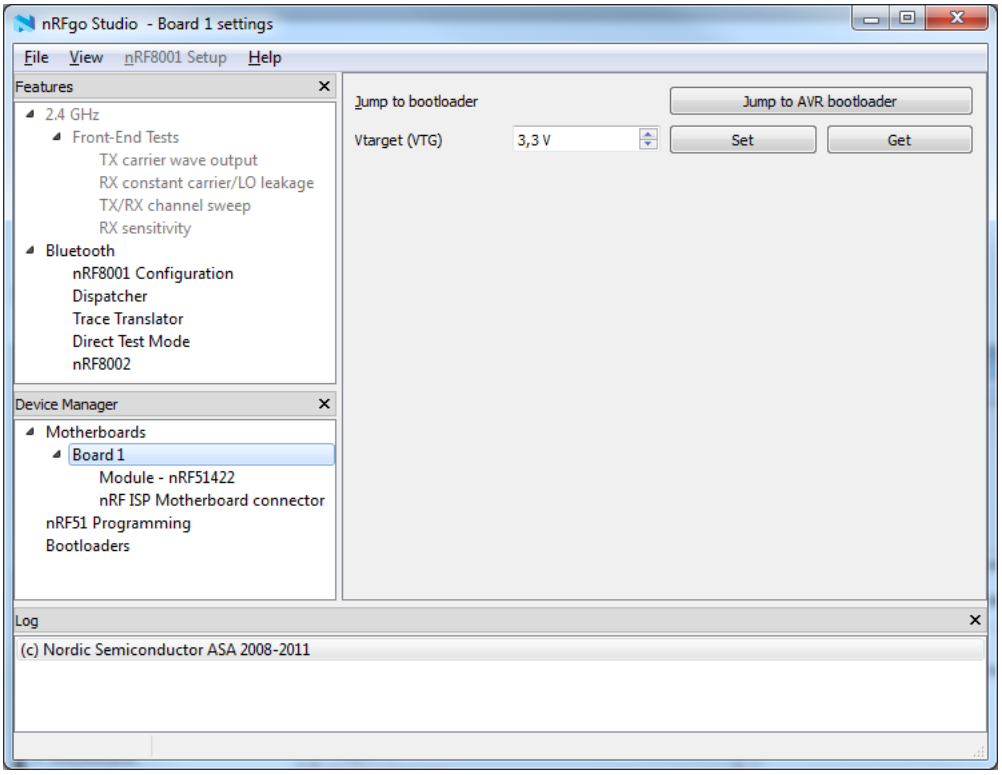
4. Start nRFgo Studio. The Motherboard firmware may require an update, and if needed, you will be prompted to begin the update. If this occurs, click **OK** and wait for the update to complete.


Upgrade firmware
The firmware on board 0 is older than the current version. It should be updated to assure correct functionality.
Do this by pressing the OK button
OK
Cancel

Note: While updating the firmware, you will see nRFgo Studio look similar to the image on the left below. On completion of the update, the "Updating…" message will change to "Upgrade" as seen in the figure on the right; however, this option is not immediately available after the latest update and will be greyed out.



5. Select the desired motherboard from the Device Manager pane in nRFgo Studio. To find the board ID, look on the 7-segment display on the Motherboard.
6. In nRFgo Studio, you can control the supply voltage (VTG) to the connected nRF51422 module. Make sure the voltage is set to 3.3 V.



Note:
- Please refer to nRFgo Studio's help file for further information. While in nRFgo Studio, press **F1** to open the help file.
- The nRF51422 device is capable of operating at voltages higher and lower than 3.3 V. This can be tested when the debugger is not required for programming or debugging the application. However, for the J-Link Lite Cortex-M (that is included in the kit) to function correctly, the supply voltage must be set to 3.3 V on the Motherboard. At all voltages other than 3.3 V, the J-Link Lite CortexM-9 debugger should be disconnected from the nRF51422 module (SEGGER has debuggers that are capable of debugging in the entire voltage range). The supply voltage operating range of the nRF51422 device is stated in the Product Specification.

Start the Blinky project

1. Locate the Blinky project found under <keil path>**\ARM\Device\Nordic\nrf51422\board\nrf6310\blinky_example\arm**.
2. Open the Blinky project in Keil µVision by double clicking the **blinky.uvproj** file.
3. Select **nRF51422** from the Select Target list and click **Build** or press **F7** to compile the Blinky project.
4. Click the **Load** icon to download and run the Blinky example firmware. **LED0** to **LED7** on the nRFgo Starter Kit Motherboard should now blink sequentially.

3.2 ANT Link demo

Download and install ANTware II

1. Go to www.nordicsemi.com and log in to your Nordic My Page account.
2. Download and install ANTware II.

Plug in the ANT dongle

1. Plug the ANT dongle (nRF2779) into your computer. The driver installation will start automatically.

Start the ANT TX broadcast example

1. Go to the ANT Broadcast TX project found under <keilpath> **\ARM\Device\Nordic\nrf51422\Board\nrf6310\ant\ant_broadcast\broadcast_tx\arm**.
2. Open the project in Keil µVision by double clicking the **ant_broadcast_tx.uvproj** file.
3. Click the **Build** icon or press **F7** to compile the project.
4. Click the **Load** icon to download and run the Broadcast example firmware.

Note: **LED0** indicates ANT is transmitting and **LED1** indicates that the device is idle.

Start ANTware II and configure the ANT channel

Below is an example of how to set the Slave setting in ANTware II.

1. Select **ANT dongle** (Device A).
2. Set Channel Assignment to **Slave**.
3. In the Channel ID window, set **Device #** to **0,** set **Device Type** to **0**, and set **Trans. Type** to **0**.
4. Click **Auto-Open**.

Monitor the traffic in the ANTware II

Received Broadcast Data is shown in the feedback window, showing packets coming from the ANT Broadcast TX. A counter is added on the two last digits and will increase by one for each packet received.

4 Development kit configuration

This chapter shows where to download third party content and how the development environment is set up.

4.1 Development environment

ARM compiler/IDE (not included in this kit)

All the source code projects and examples can be compiled and used with the Keil Microcontroller Development Kit (MDK). For full use of the Development Kit source code projects, and to upgrade firmware, download and install the free KEIL MDK-ARM Lite from https://www.keil.com/demo/eval/arm.htm.

J-Link Lite CortexM-9 driver (not included in this kit)

For installing drivers for the SEGGER J-Link Lite CortexM-9, visit www.segger.com and go to the **Downloads** section. Select the J-Link Lite and download the software and documentation. You must correctly install the drivers for the device to use the J-Link debugger with Keil MDK. See ***Appendix A ": Installing drivers and configuring KEIL projects for the SEGGER debugger" on page 45.***

4.1.1 Development environment setup

The nRF51422 device can be programmed from several environments. This section shows the development setup using Keil MDK-ARM.

The nRF51422 DK can be configured for end-to-end ANT applications or for standard ANT profile-based applications, depending on your development requirements.

For development of end-to-end ANT applications, you need the following:

- **2 x** Motherboards (from the nRFgo Starter Kit - not included)
- **2 x** nRF51422 modules

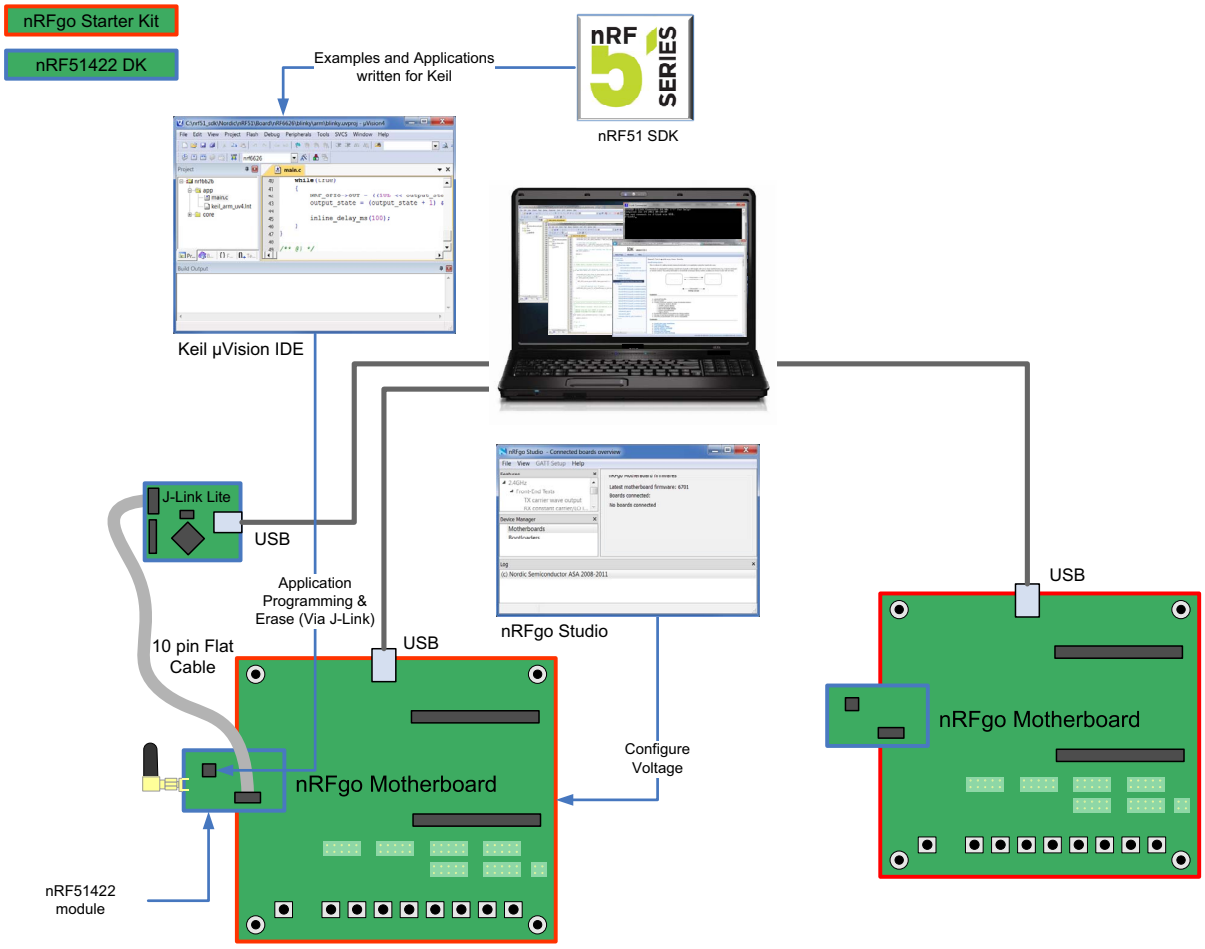


Figure 2 *nRF51422 DK configuration for end-to-end ANT development platforms*

For development/demonstration of standard ANT profile-based applications, you need the following:

- **1 x** Motherboard with attached nRF51422 module
- **1 x** nRF2779 (AP2) dongle
- ANTware II PC software

The relationship between the hardware and software components, and the Motherboard(s) can be seen in ***Figure 2 on page 13*** and ***Figure 3 on page 14***.

Note: The Keil µVision IDE is not included in the kit content.

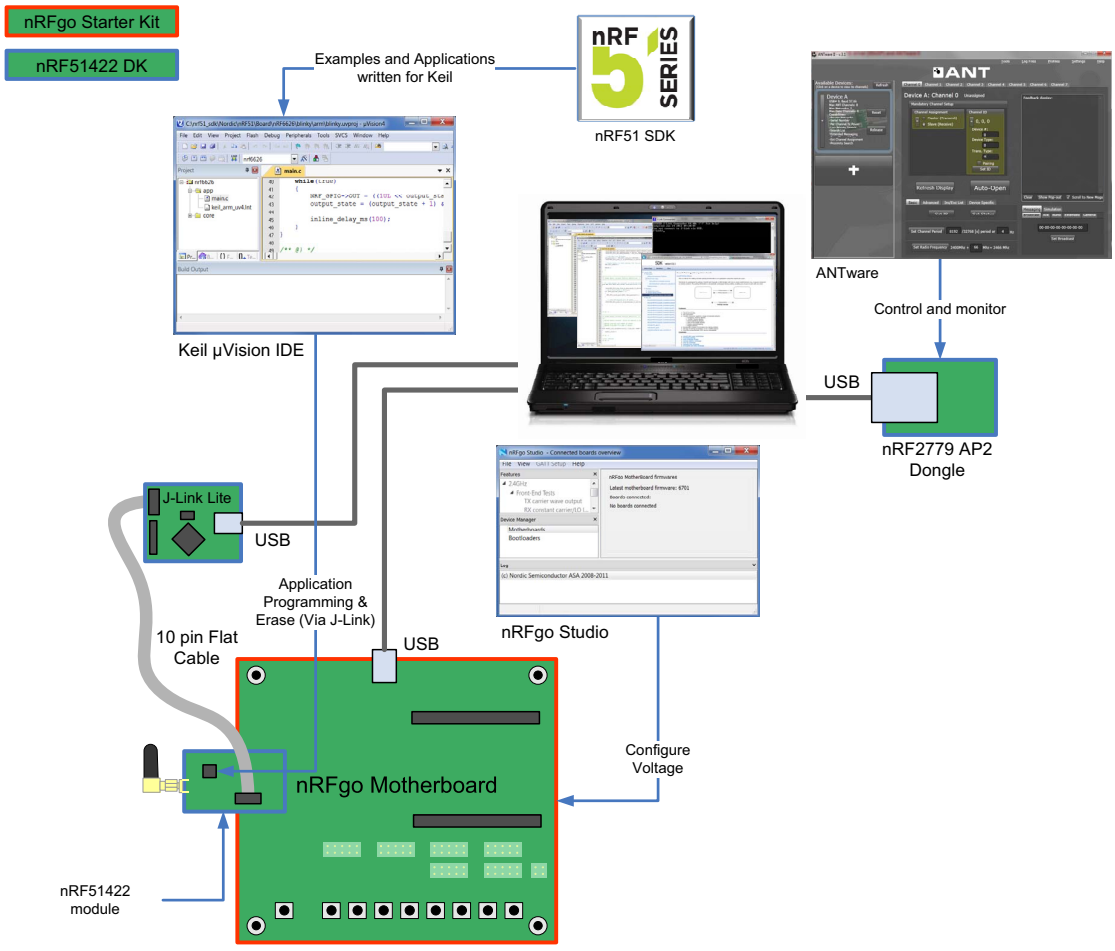


Figure 3 *nRF51422 DK configuration for one ANT development platform*

4.2 Kit setup

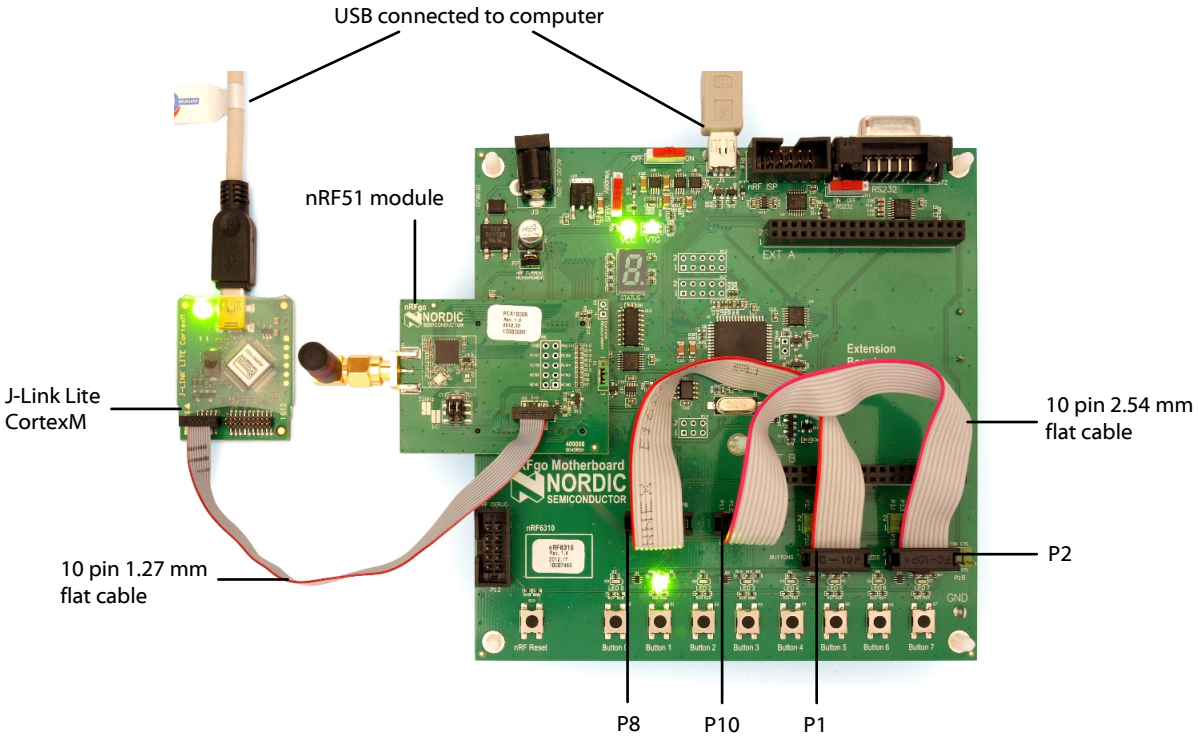


Figure 4 *Kit setup*

4.2.1 nRFgo nRF51422 DK modules

The nRF51422 modules contain the nRF51422 device and its interfaces (see ***section 5.1 on page 17*** for a complete description). The modules are plugged into the Motherboard which provides the power supply. Make sure the jumper on header **K1** is placed in the default position as described in ***section 5.1.8 on page 23***. The Motherboard is connected to your computer with a USB cable and turned on as described in ***"Turn on and set the supply voltage" on page 8***.

On the Motherboard, the two 10 pin flat cables connecting **P8** (PORT0) and **P1** (BUTTONS), and **P10** (PORT1) and **P2** (LEDS) are required to connect the correct I/Os to the LEDs for the example projects in ***chapter 2 on page 4*** and for projects in the SDK.

Note: With the nRFgo Studio application you can control the supply voltage (VTG) to the connected nRF51422 DK module as described in ***"Turn on and set the supply voltage" on page 8.***

4.2.2 nRFready ANT USB dongle (nRF2779)

The nRFready ANT USB dongle (nRF2779) enables you to send data to nRF51422 as well as monitor the data sent from nRF51422. The nRFready ANT USB dongle is plugged into a USB port on your computer as described in ***"Plug in the ANT dongle" on page 10***.

4.2.3 J-Link Lite CortexM-9 JTAG/SWD Emulator

The programming and debugging (SWD) interface of the nRF51422 device is accessed through a 10 pin connector (**P3**) on the nRF51422 DK module.

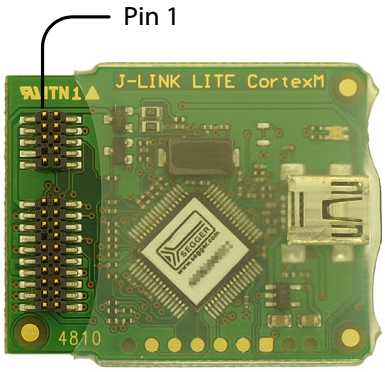


Figure 5 *Pin 1 position on the SEGGER J-Link Lite CortexM-9*

Connect the JTAG/SWD emulator using the 10 pin 1.27 mm flat cable supplied with the kit to the nRF51422 DK module at **P3** and to your computer with a USB cable, as shown in ***Figure 6***.

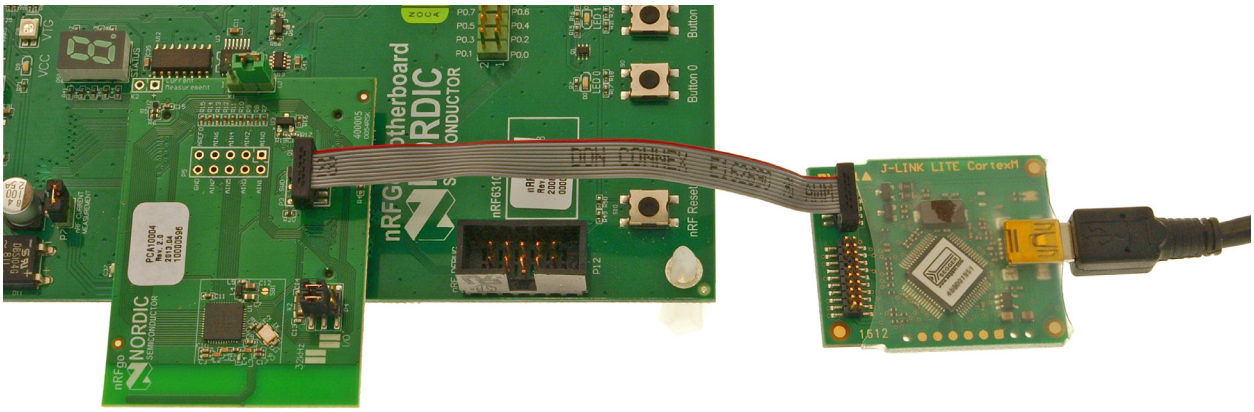


Figure 6 *nRF51422 DK module connected to the SEGGER J-Link Lite CortexM-9*

5 Hardware description

This chapter describes the nRF51422 DK modules and the nRFready ANT USB dongle (nRF2779). For driver downloads, please register your Development Kit in your My Page account to access the files.

5.1 nRFgo nRF51422 DK modules

The nRF51422 modules (PCA10006, PCA10007) are delivered with the S210 nRF51422 SoftDevice (ANT) preloaded on the nRF51422 chip. It is not possible to erase or upgrade the SoftDevice on this chip.

5.1.1 Key features

The nRF51422 DK modules have the following key features:

- nRF51422 flash based ANT™ SoC solution
- Current Shunt Monitor (CSM) for current measurements
- nRFgo Motherboard integration
- PCB antenna (PCA10006 only)
- SMA connector compatibility (PCA10007 only)
- SWD interface connector for programming and debugging

5.1.2 Hardware pictures

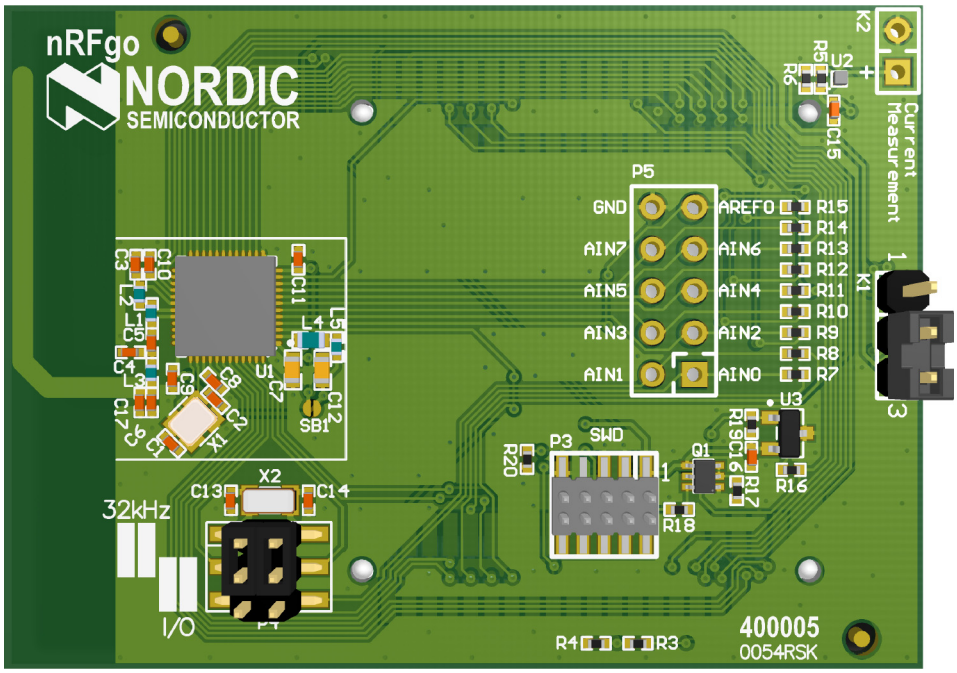


Figure 7 *nRF51422 DK module (PCA10006) top*

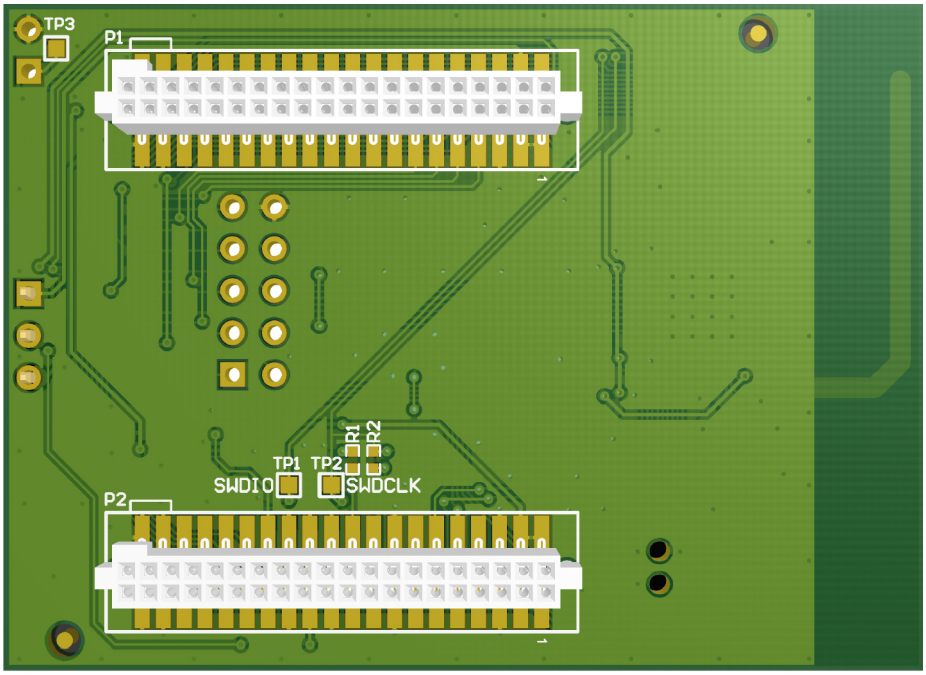


Figure 8 *nRF51422 DK module (PCA10006) bottom*

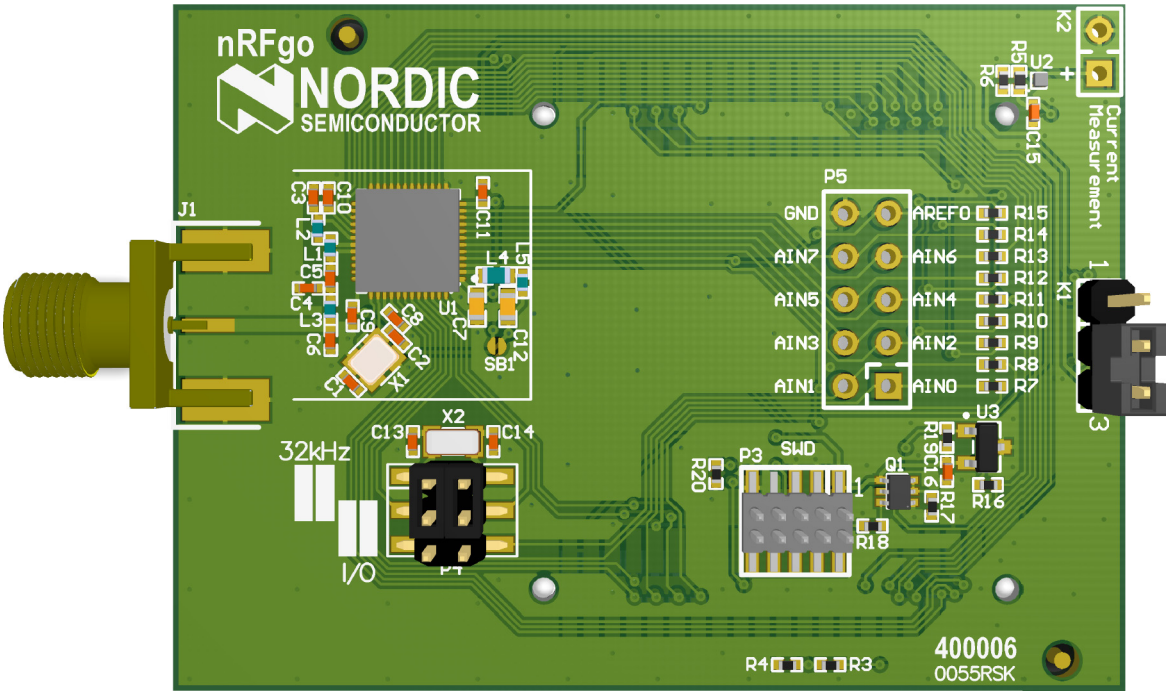


Figure 9 *nRF51422 DK module (PCA10007) top*

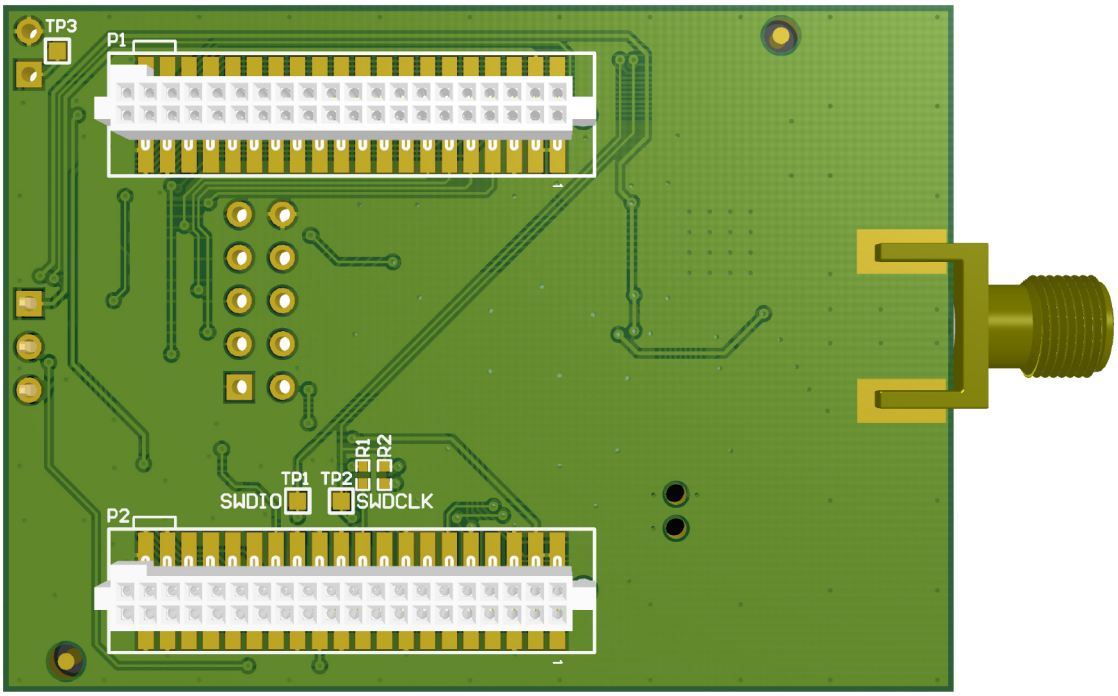


Figure 10 *nRF51422 DK module (PCA10007) bottom*

5.1.3 Block diagram

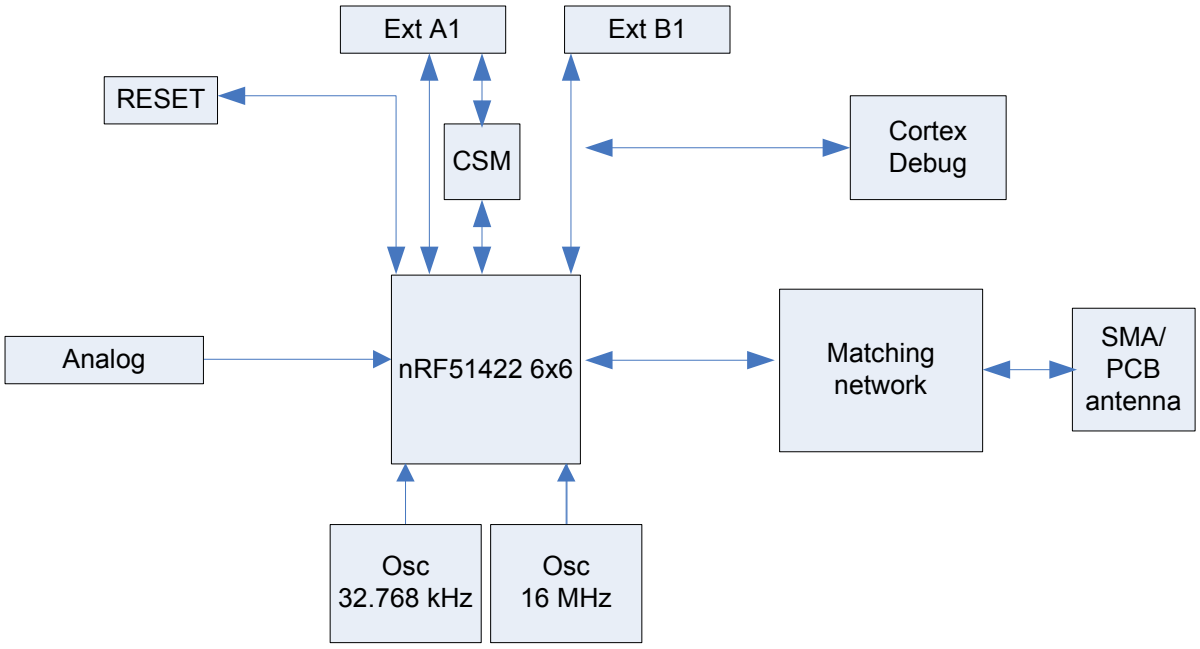


Figure 11 *nRF51422 DK module block diagram*

5.1.4 nRFgo nRF51422 DK module extension

Connectors **P1** and **P2** on the nRF51422 DK module connect to the Motherboard.

VCC	1	2	VCC_nRF'	VTG	1	2	VTG
VCC	3	4	VCC_nRF'	VTG	3	4	VTG
GND	5	6	GND	GND	5	6	GND
P0.00	7	8	P0.01	P0.24	7	8	P0.25
P0.02	9	10	P0.03	P0.26	9	10	P0.27
P0.04	11	12	P0.05	P0.28	11	12	P0.29
P0.06	13	14	P0.07	P0.30	13	14	Not in use
GND	15	16	GND	GND	15	16	GND
Not in use	17	18	Not in use	Not in use	17	18	Not in use
Not in use	19	20	Not in use	Not in use	19	20	Not in use
GND	21	22	GND	GND	21	22	GND
Not in use	23	24	Not in use	BoardID	23	24	GND
Not in use	25	26	RESET	GND	25	26	GND
Not in use	27	28	Not in use	Not in use	27	28	BoardID_EE
P0.08	29	30	P0.09	P0.16	29	30	P0.17
P0.10	31	32	P0.11	P0.18	31	32	P0.19
P0.12	33	34	P0.13	P0.20	33	34	P0.21
P0.14	35	36	P0.15	P0.22	35	36	P0.23
GND	37	38	GND	GND	37	38	GND
GND	39	40	GND	GND	39	40	GND
	P1				P2		

Figure 12 *nRF51422 DK module connectors - **P1** and **P2***

5.1.5 Analog inputs

Direct access to the nRF51422 analog input is available on connector **P5** on the nRF51422 DK module.

Note: **P5** is not mounted on the board.

The analog inputs are routed through the extension connectors to the Motherboard. To avoid noise from the Motherboard, the 0 Ω resistors must be removed.

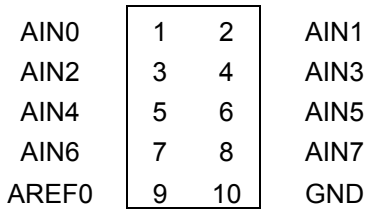


Figure 13 *Analog inputs connector **P5** on the nRF51422 DK board*

5.1.6 SWD interface

The programming and debugging interface is accessed through the 10 pin connector **P3** on the nRF51422 DK module.

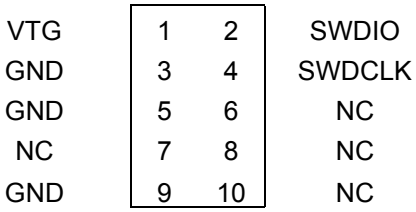


Figure 14 *SWD interface connector P3*

Pin	Label	Description
1	VTG	Reference voltage for programmer
2	SWDIO	Serial Wire Debug Data
3	GND	Ground
4	SWDCLK	Serial Wire Debug Clock
5	GND	Ground
6	NC	No Connection
7	NC	No Connection
8	NC	No Connection
9	GND	Ground
10	NC	No Connection

Table 1 *SWD interface connector pin description*

5.1.7 32.768 kHz crystal

The nRF51422 can use an optional 32.768 kHz crystal (X2) for higher accuracy and lower average power consumption. On the nRF51422 DK module, P0.26 and P0.27 are disconnected from the Motherboard and connected to the two jumpers on the nRF51422 DK module at the position 32 kHz on connector **P4** (Pin 1 and Pin 3 for XL1 and Pin 2 and Pin 4 for XL2). This is the default position of the jumpers when shipped.

If the jumpers are placed in the position I/O on connector **P4** (Pin 3 and Pin 5 for XL1 and Pin 4 and Pin 6 for XL2), P0.26 and P0.27 connects to the Motherboard and can be used for normal I/O.

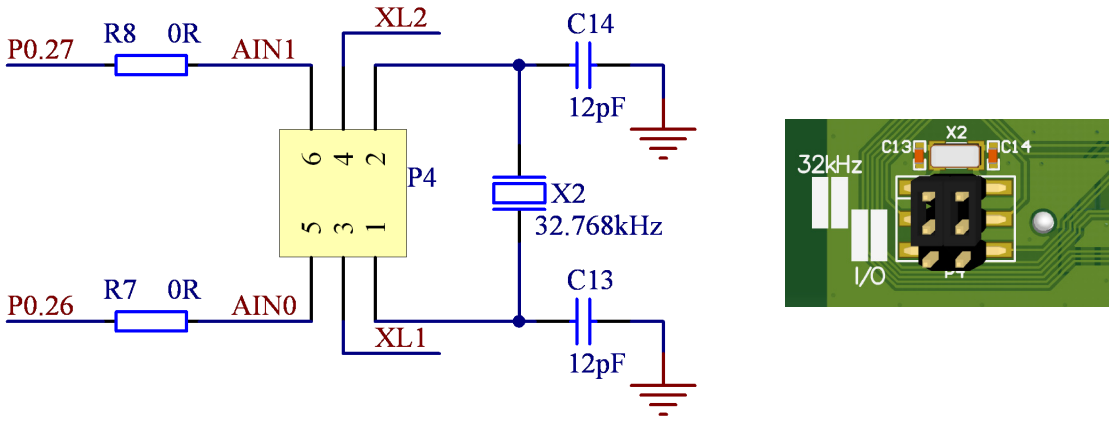


Figure 15 *32.768 kHz crystal circuit schematic and PCB*

5.1.8 Current measurements

The current drawn from the nRF51422 device can be monitored using the Current Shunt Monitor (CSM), INA216 (U2). The gain of the CSM is set to 200 V/V for lowest possible drop voltage.

For current measurements using the CSM, the jumper on connector **K1** must be placed on Pin 2 and Pin 3. This is the default position of the jumpers when shipped. Current on the nRF51422 module is measured on connector **K2** as voltage proportional to the current.

For current measurements using the jumper **P7** on the Motherboard, the jumper on connector **K1** must be placed on Pin 1 and Pin 2.

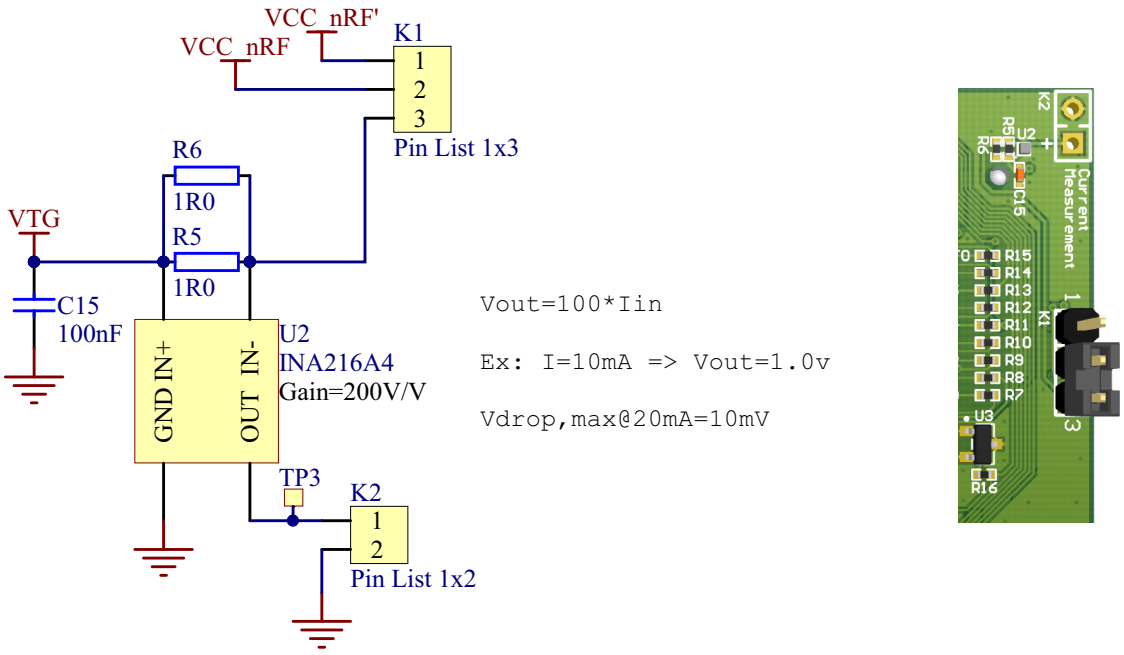


Figure 16 *Current measurement circuit schematic and PCB*

5.2 nRFready ANT USB dongle (nRF2779)

The nRFready ANT USB dongle (nRF2779) enables you to send data to nRF51422 as well as monitor the data sent from nRF51422. Software and drivers for the ANT USB dongle install automatically, but are also available for download from the nRFready ANT USB dongle product page at www.nordicsemi.com. The ANT USB dongle combined with ANTware II gives you a peer device for nRF51422 that you can use to test the wireless connection.

5.2.1 Key features

The nRF2779 has the following key features:

- nRF24AP2 IC
- ANT™ compatible

5.2.2 Hardware pictures

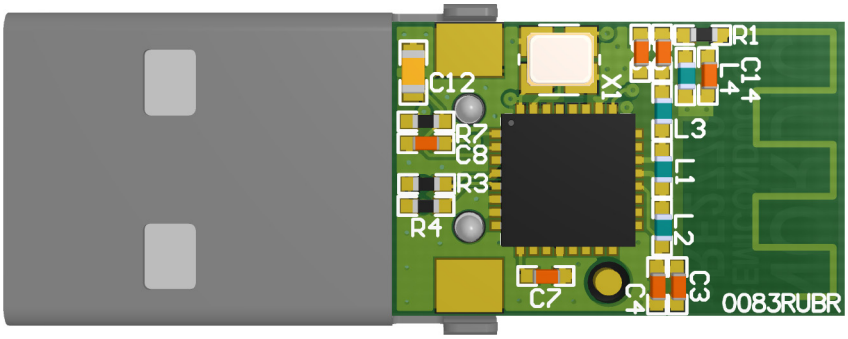


Figure 17 *nRF2779 top side*

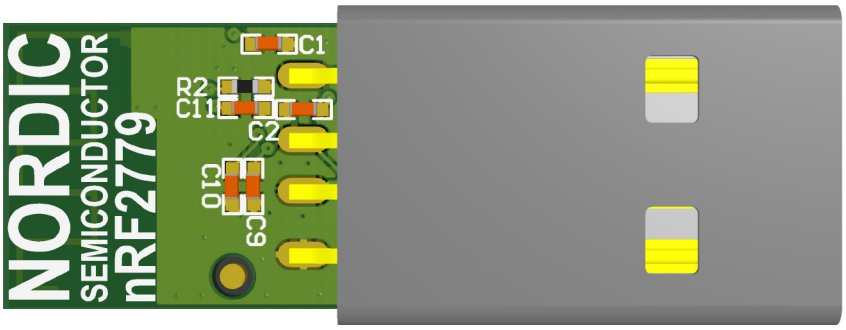


Figure 18 *nRF2779 bottom side*

5.2.3 Block diagram

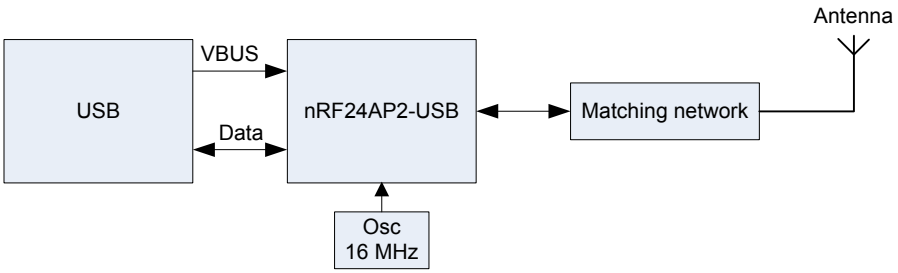


Figure 19 *nRF2779 block diagram*

6 Flash programming and application development

The nRF51422 chip is shipped with the SoftDevice pre-programmed on the chip. It is not possible to upgrade or erase the SoftDevice on nRF51422.

6.1 Programming and erasing flash using nRFgo Studio

Use nRFgo Studio to program or erase an application HEX file onto the nRF51422 chip.

Note: For details on memory organization and protection see the *nRF51 Series Reference Manual*.

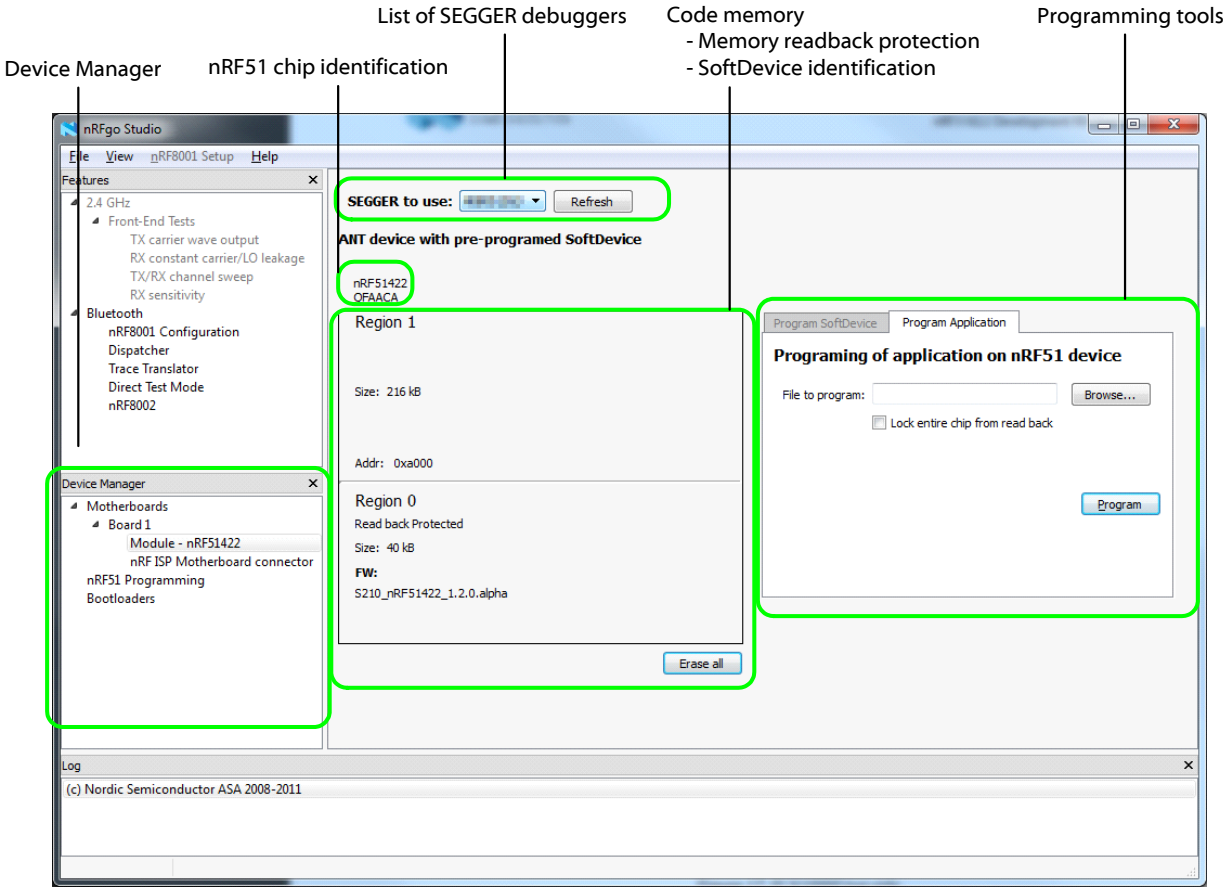


Figure 20 *nRFgo Studio dashboard*

6.1.1 Selecting a board to program

1. Open nRFgo Studio.
2. In the Device Manager pane select which board to program or erase.
3. The nRF51422 DK modules (PCA10006/ PCA10007) cannot be selected directly. The boards must be selected through the J-Link debugger connected to them. To do this, you either select the actual module located under **Motherboard > Boardx** or by clicking on **nRF51 Programming**.
4. Select the debugger you want to use from the list of available J-Link debuggers under **Segger to use**.

6.1.2 Identifying the nRF51 chip and chip content

When you select a board, nRFgo Studio identifies the nRF51422 chip and how its memory is organized. The following chip and memory information is displayed:

- **nRF51 chip identification** - Identifies the chip by name and code variant (for example, nRF51422 QFAACA). If the debugger is not connected to the chip, or the debugger has a problem communicating with the chip, it will show the following message "No device detected. Ensure that you have the SEGGER connected correctly to the board and that the board is powered and configured for debugging."
- **Code memory -** Shows how the code memory is organized in one or two regions (Region 0 and 1) and the size of each region. nRF51422 will always show two regions. For devices containing a SoftDevice, the code memory is divided in two regions, with the SoftDevice in Region 0. The tool shows you how much memory is used by the SoftDevice and how much is left for the application.
  - Memory readback protection - Shows how the readback protection is set. The two possible options are readback protection on Region 0 or readback protection of the whole code memory.
  - SoftDevice identification - nRFgo Studio tries to identify the firmware located in the chip at Region 0. For the firmware that it recognizes it prints the ID (in clear text) for the unrecognized firmware it prints the FWID number.

6.1.3 Erase all

Use Erase all in the following situation:

- You have programmed an application on top of the S210 nRF51422 SoftDevice and selected **Lock entire chip from read back**. Once you have performed Erase All, the application will be erased from the chip. The S210 nRF51422 SoftDevice will not be erased.

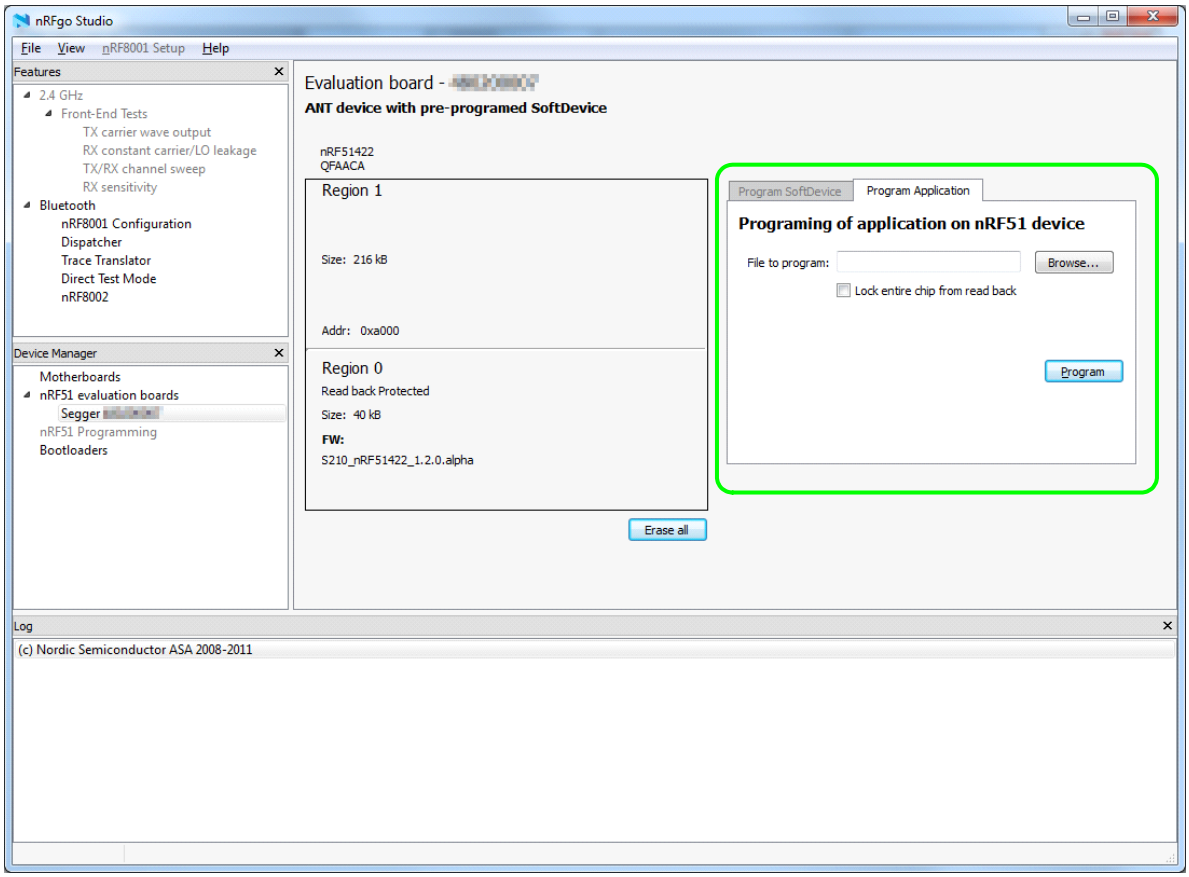
To use the **Erase all** function, follow the steps in ***section 6.1.1 on page 27***. Then click **Erase all**.

6.1.4 Programming an application

This function lets you program an application onto the chip (on top of the S210 nRF51422 SoftDevice).

Before nRFgo Studio starts programming it verifies that the HEX file matches the actual memory configuration. If it matches, nRFgo Studio continues with the programming, if not it stops the programming and returns an error message.

This programming will not set up any memory Regions.



1. Follow the steps in ***section 6.1.1 on page 27*** and then select the **Program Application** tab.
2. Click **Browse** and select the HEX file to program.
3. Select whether to enable or disable readback protection of the entire chip. If you enable readback protection, you will have to do an Erase All to reprogram the chip again.

Note: A chip that is programmed with **Lock entire chip from read back** enabled will not work with a development toolchain. To make it work you must perform Erase all.
Lock entire chip from read back can be used to prevent an accidental overwrite of the chip content.

6.2 Application development

The user application is compiled, linked, and downloaded independently from the SoftDevice. This means that developing and debugging on a chip pre-programmed with a SoftDevice is similar to that of a blank chip. The main differences are memory layout and the call stack size.

6.2.1 Configuring memory layout

Specific SoftDevice versions and stacks could have different requirements. Please review these before proceeding.

The applications vector table must be set up differently depending on whether it will run on a chip that is blank or pre-programmed with a SoftDevice.

The SoftDevice program area starts at address 0x0 and has a predefined size. The application start vector must be placed right after the SoftDevice. The available size has to be set so that it uses the remaining memory for the application. Similarly, the SoftDevice data area starts at the lowest RAM address. The application data area must be placed after the SoftDevice data area.

Table 2 shows examples for setting up the start address and size depending on the code and data size used by the SoftDevice. The example is based on a chip with 256 kB of code memory and 16 kB of RAM.

Device configuration	SoftDevice		App. code start address	Available code memory	App. data start address	Available RAM
	Code memory usage	RAM usage				
Blank chip	0 kB	0 kB	0x0	0x40000	0x20000000	0x4000
SoftDevice A	64 kB	2 kB	0x10000	0x30000	0x20000800	0x3800
SoftDevice B	128 kB	8 kB	0x20000	0x20000	0x20002000	0x2000

Table 2 *SoftDevice memory layout*

Note: See the *nRF51422 Product Specification* for details on the total code memory and RAM available in the device. The amount of code memory and RAM used by the SoftDevice is described in the *S210 nRF51422 SoftDevice Specification*.

There are two ways to configure the memory layout:

- Using Keil IDE
- Using a Scatter file (not covered in this document)

Note: The example code given by Nordic Semiconductor configures the memory layout in the Keil IDE. Scatter file loading is not available when using the evaluation version of Keil IDE.

6.2.1.1 Memory layout configuration in Keil IDE

To access the Keil IDE memory layout:

1. Click the **Project** menu and select **Options for Target.**
2. Select the **Linker** tab.
3. Check **Use memory layout from Target Dialog**.

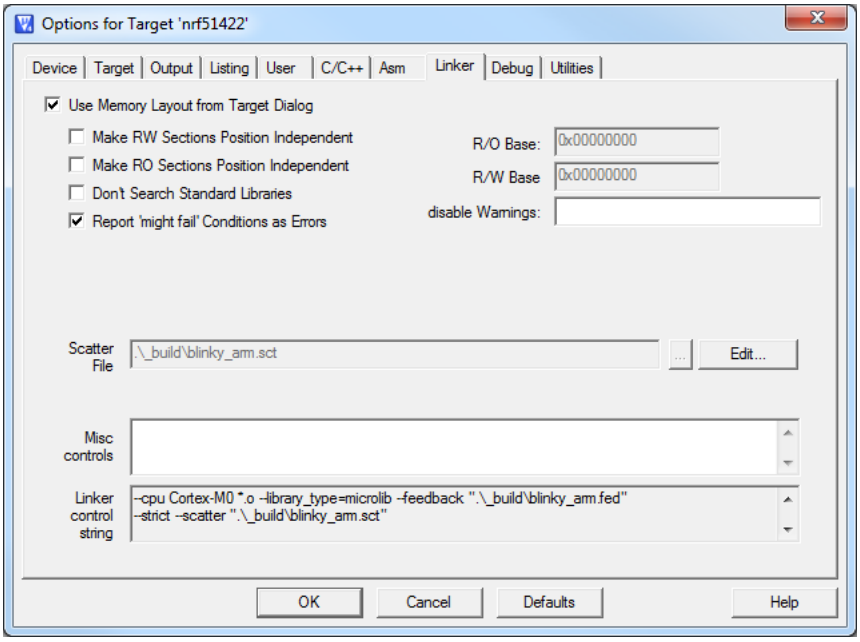


Figure 21 *Keil linker settings*

4. Select the **Target** tab.
5. In **Read/Only Memory Areas**, define values for **Start** and **Size**.
6. In **Read/Write Memory Areas**, define values for **Start** and **Size** as seen in ***Figure 22***.
7. Click **OK**.

Below is an example configuration for an application using a chip with 256 kB of code memory and 16 kB of RAM, and a SoftDevice using 128 kB of code memory and 8 kB of RAM (SoftDevice B described in ***Table 2 on page 29***).

- Base code memory address 0x20000 and available code memory size is 0x20000 (128 kB).
- Base RAM memory address 0x20002000 and available RAM size is 0x2000 (8 kB).

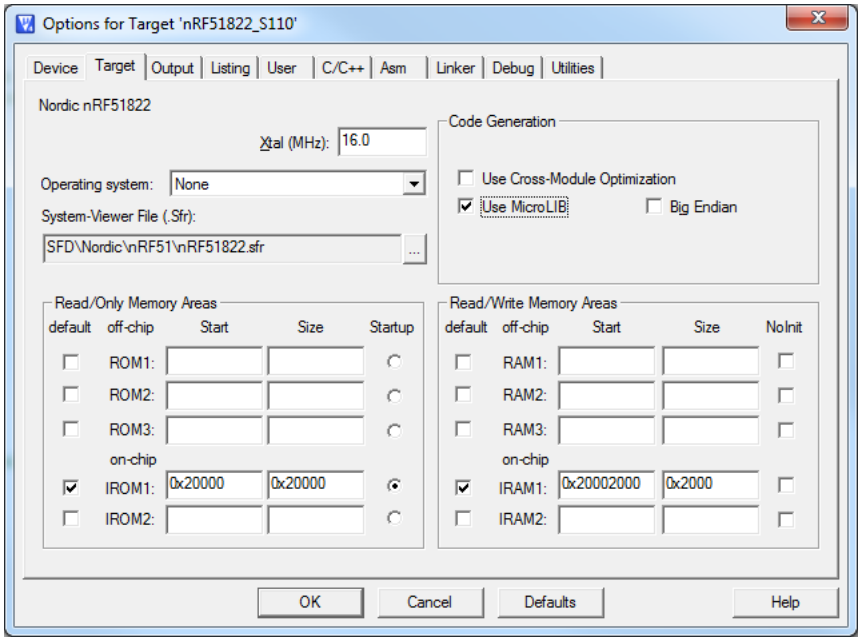


Figure 22 *Memory layout with example SoftDevice*

Memory		Description
IROM1	Start	Specify the start address for the application code
	Size	Specify available code size for the application code
IRAM1	Start	Specify start address for the application data
	Size	Specify available RAM size for the application data

Table 3 *Memory layout*

6.2.2 Shared call stack

The user application shares the call stack with the SoftDevice if the SoftDevice is loaded on the chip. The application must reserve enough memory for both itself and the SoftDevice in the call stack. The call stack size required by the SoftDevice varies between devices and protocol stack versions, and is supplied in the *S210 nRF51422 SoftDevice Specification*.

The user application sets its call stack size plus the amount needed by the SoftDevice. It then writes the stack pointer at the first address of the application Reset Vector.

Note: Using Keil with the ARMCC toolchain, the call-stack size can be set using the `Stack_Size` definitions in your projects startup file, typically `arm_startup_nrf51.s`.

```
Stack_Size      EQU     0x400 ; The application call-stack size + protocol
call-stack size
                AREA    STACK, NOINIT, READWRITE, ALIGN=3
Stack_Mem       SPACE   Stack_Size
__initial_sp
```

6.2.3 Debugger configuration

Project files delivered in the SDK are configured and ready for download and debugging. If a new application project is used, the debugger must be properly configured. To configure the debugger:

1. In Keil, select **Options for Target** (**ALT+F7**) from the **Project** menu. The **Options for Target** dialog box appears.
2. Select the **Debug** tab.
3. Apply the **Use** option and select the J-Link/J-Trace debugger from the list.
4. Set **Driver DLL** to SARMCM3.DLL.
5. Set **Dialog DLL** to TARMCM1.DLL.

Other options can be selected as needed. To take full advantage of the debugger and its features, the following are advised:

- Breakpoints
- Load Application at Startup
- Memory Display
- Toolbox
- Watch Windows

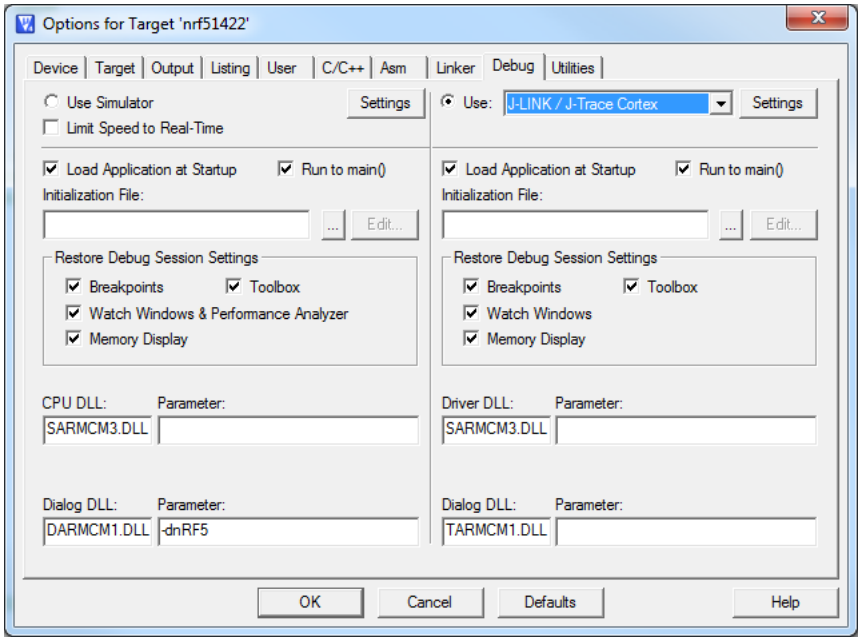


Figure 23 *Debugger options*

6. Click **Settings** next to the **Use** field in the top right of the window.

7. In the Target driver setup, provide information about debugging protocol and maximum speed. Select **SW** in the **Port** drop-down.
8. In **Max Clock** the maximum speed for the debugging port cannot be exceeded (1 MHz). A proper configuration is shown in ***Figure 24 on page 34***.

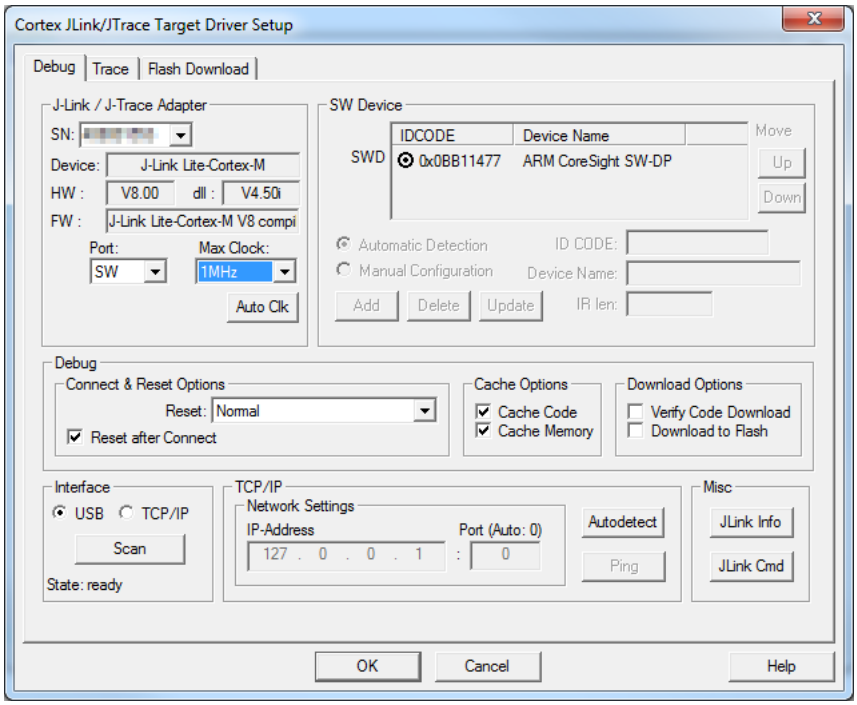


Figure 24 *Target driver setup*

Click **Start/Stop Debug Session** (**CTRL+F5**) in the Keil IDE to start debugging.

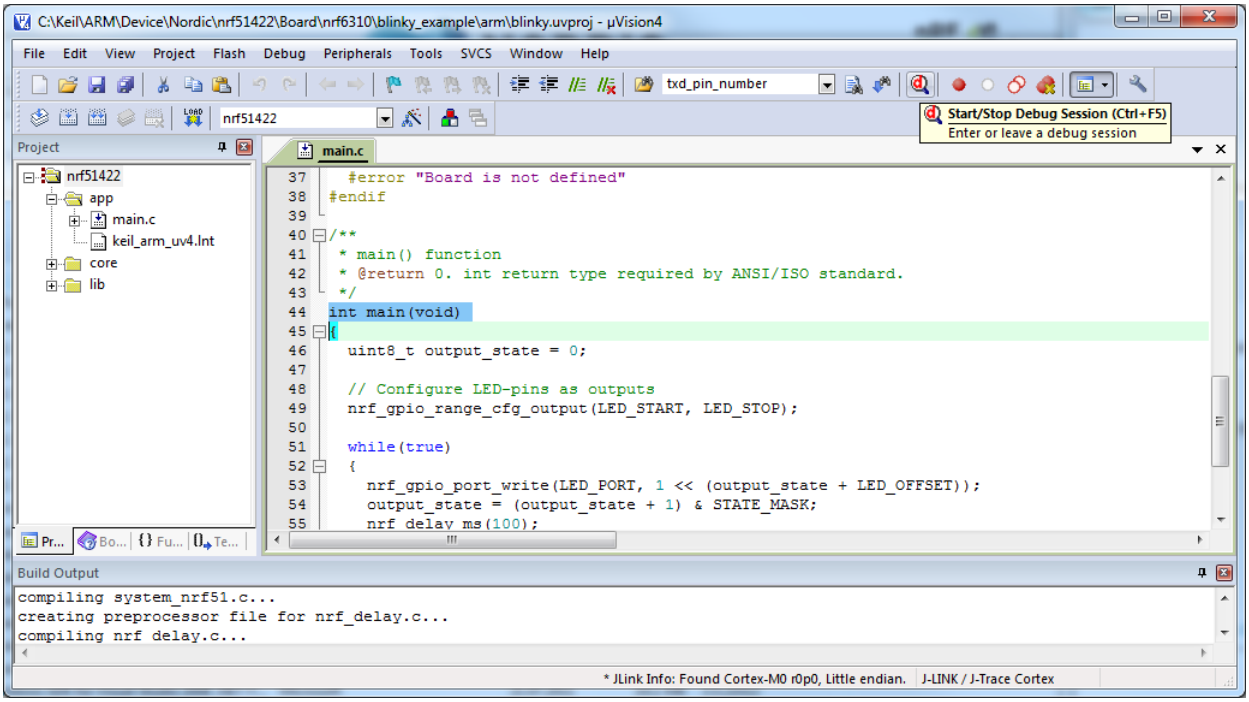


Figure 25 *Debugger initiation*

6.2.4 Limitation when debugging on a chip with a SoftDevice

When a SoftDevice is installed in a device, there are certain limitations when debugging.

The nRF51422 has a Memory Protection Unit that prevents the debugger from having read/write access to the SoftDevice code memory area. The debugger reads this area as 0x0000 (no operation instruction).

6.2.5 Programming the device

The code memory area occupied by the SoftDevice is read, write, and erase protected. The debugger will read this area as 0x0000 (no operation instruction). When the SoftDevice is enabled, the Memory Protection Unit enables write protection for certain peripherals used by the protocol stack. Protected peripherals are described in the *S210 nRF51422 SoftDevice Specification*.

Configure the debugger and compile and link the application code. Download the application using the Keil IDE download button. To configure and start the download:

1. Select **Options for Target** in the **Project** menu.
2. Select the **Utilities** tab in the **Options for Target** dialog box.
3. Click **Settings**.
4. Select the **Program** check box.
5. Choose **Erase Full Chip**.
6. Click **Add** and select the nRF51xxx algorithm from the list to select the programming algorithm used by Keil IDE.
7. Click **Download**.

Note: The nRF51xxx algorithm is installed automatically during the SDK installation. This algorithm is a generic nRF51 series algorithm, which provides download capabilities to all series devices up to 2 MB of code memory.

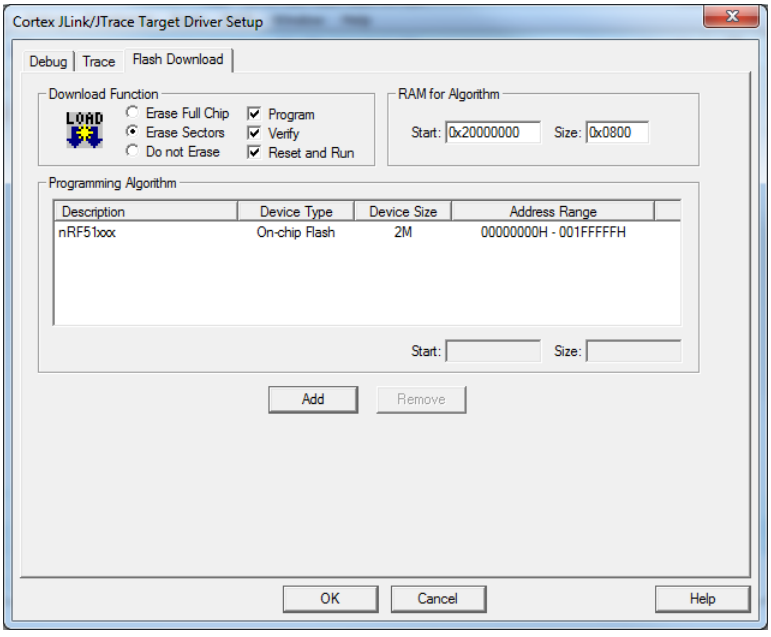


Figure 26 *Selecting J-Link/J-Trace Cortex*

8. In the **Utilities** tab, select **Use Target Driver for Flash Programming**.

Choose the available debugger from the list as shown in ***Figure 27***.

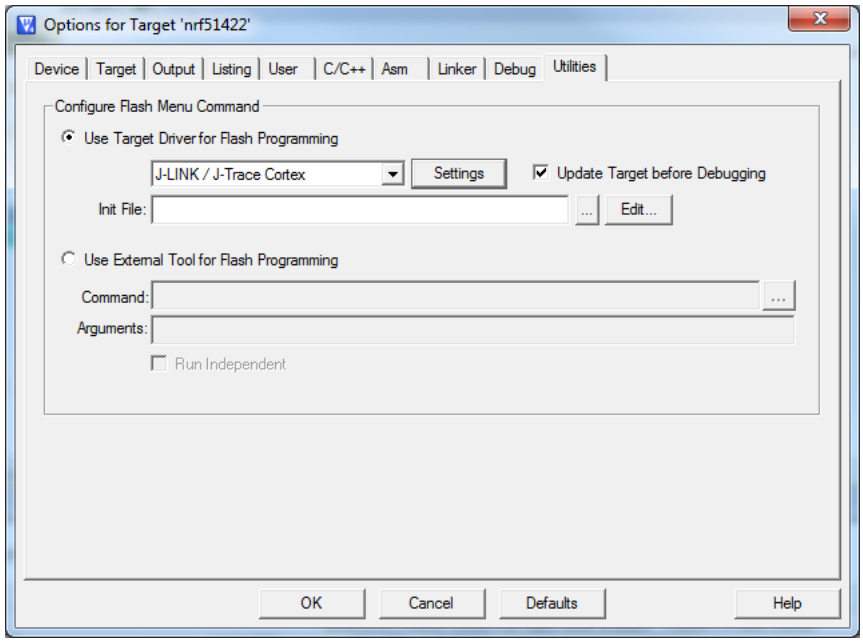


Figure 27 *Debugger selection*

6.2.6 Erasing the device

The code memory area available for the user application can be erased by doing the following:

1. In the Target options dialog select the **Utilities** tab.
2. Click **Settings** and select **Erase Full Chip**.
3. Uncheck **Program** and **Verify**.

A normal download procedure will erase the device application code memory area.

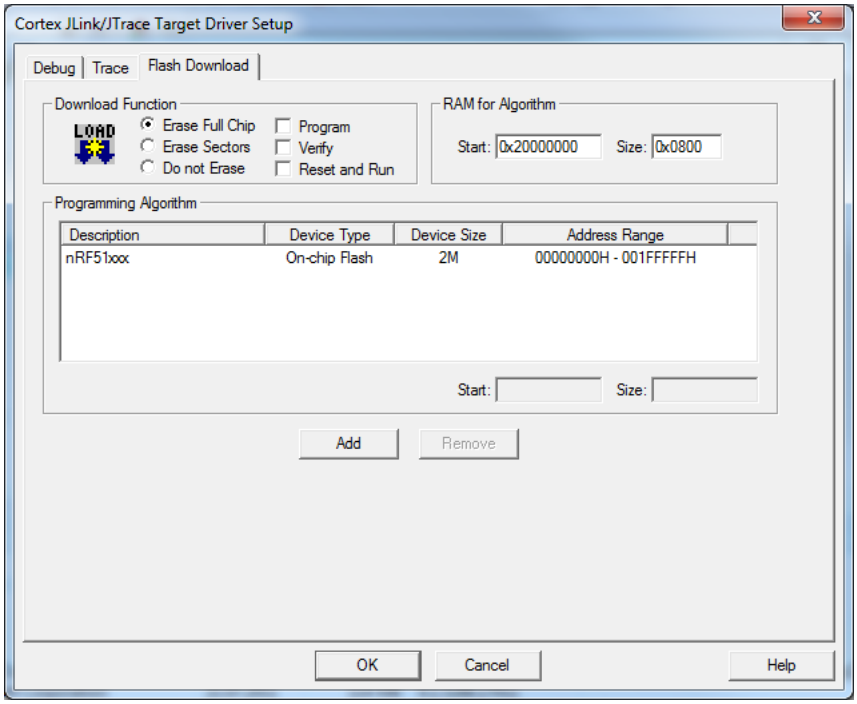


Figure 28 *Erasing*

Note: The SoftDevice pre-programmed in the device cannot be erased. Any erase procedure will only remove the information in the code memory area available for the user application.

7 Debugging the nRF51422

For debugging with SEGGER J-Link, see ***Appendix A on page 45***. For general information of how to debug using the Keil µVision IDE, see http://www.keil.com/uvision/debug.asp.

The following steps tell how to configure the debugger in Keil.

1. In the **Project** menu click **Options for Target** (**CTRL+F7**).
2. Select the **Debug** tab.
3. To enter debugging mode click **Start/Stop Debug Session** or **CTRL**+**F5**.

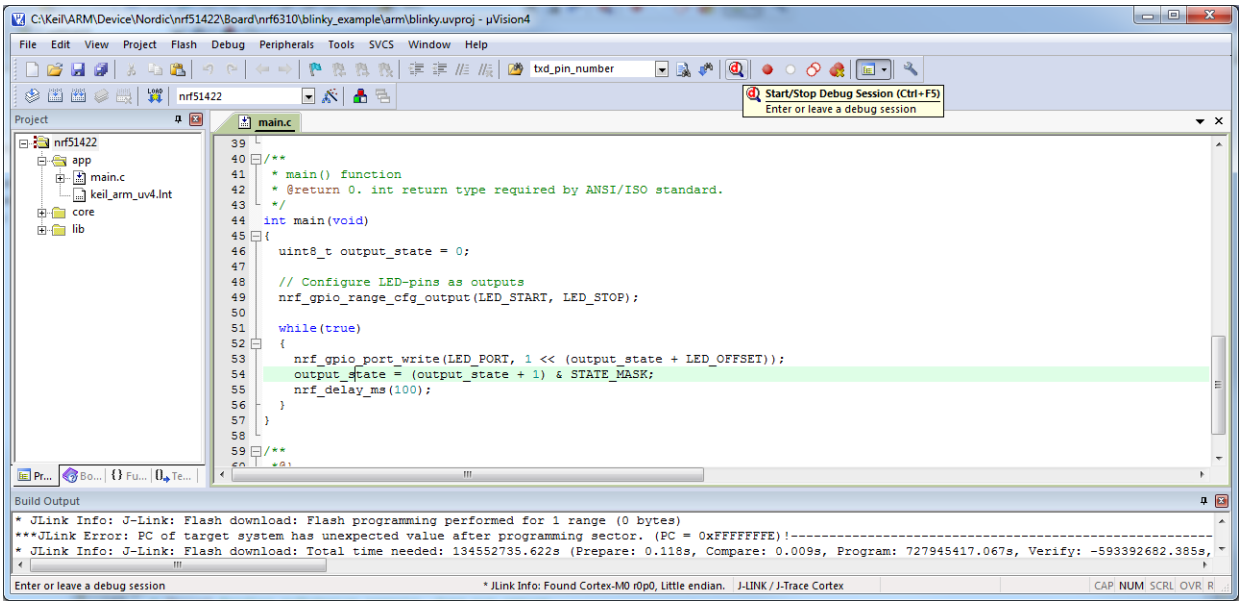


Figure 29 *Start debugging mode*

7.1 nRF51422 debug features and precautions

This section contains information about the System Viewer Windows and debugging an application when a readback protected SoftDevice is present.

7.1.1 System Viewer windows

The System Viewer enables you to select device peripherals and see their contents in separate windows. In the **View** menu point to **System Viewer** and select the peripheral you want to see. The peripheral register values are displayed in their respective pane in Keil, as seen in ***Figure 30***. More information on System Viewer can be found at: http://www.keil.com/uvision/db_view_sysview.asp.

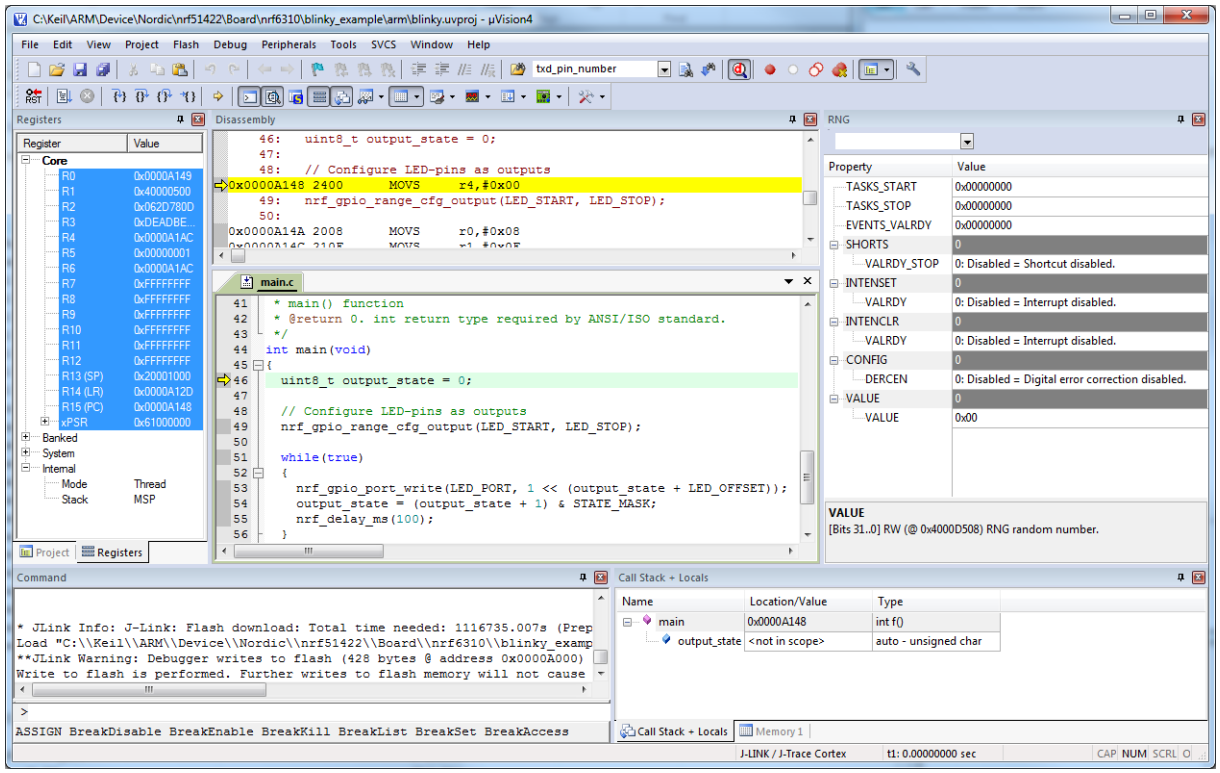


Figure 30 *System Viewer window of the RNG peripheral*

7.1.2 Debugging an application when a readback protected SoftDevice is present

Debugging applications with a SoftDevice present behaves as described in http://www.keil.com/uvision/debug.asp, except when the program counter is in Region 0 on a SoftDevice with readback protection enabled. Code words from addresses in the protected area will always return zero to the debugger.

Any values in peripheral registers that are restricted or blocked by the SoftDevice will be invisible to the debugger as well. Information on the SoftDevice configuration and memory resource mapping can be found in the *S210 nRF51422 SoftDevice Specification*.

Note: Avoid single stepping to the protected area. Instead, set the breakpoint right after SVC calls while debugging and run the application to the actual breakpoint, see ***Figure 32 on page 41***. The "step over" function (**F10**) may also be used instead to step over SVC calls to avoid delays when entering the readback protected area.

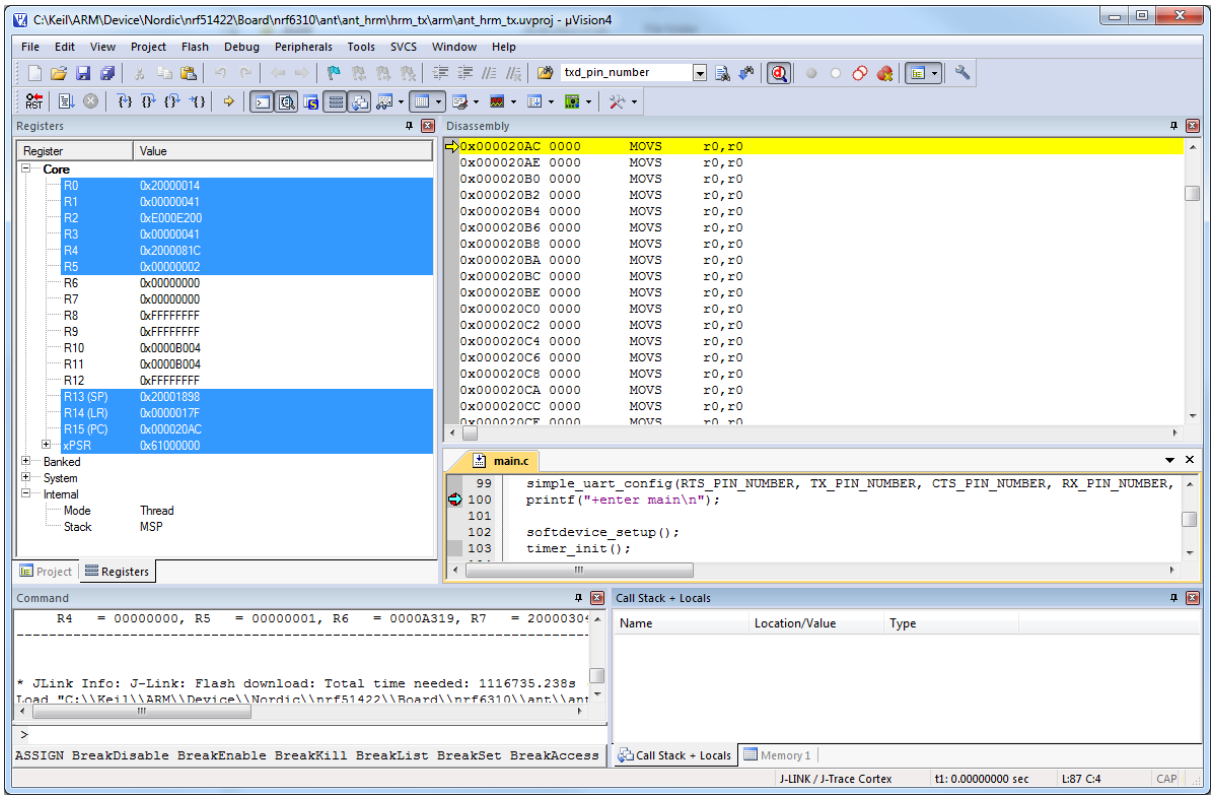


Figure 31 *Debugger information for a setup with a SoftDevice enabled in the protected area*

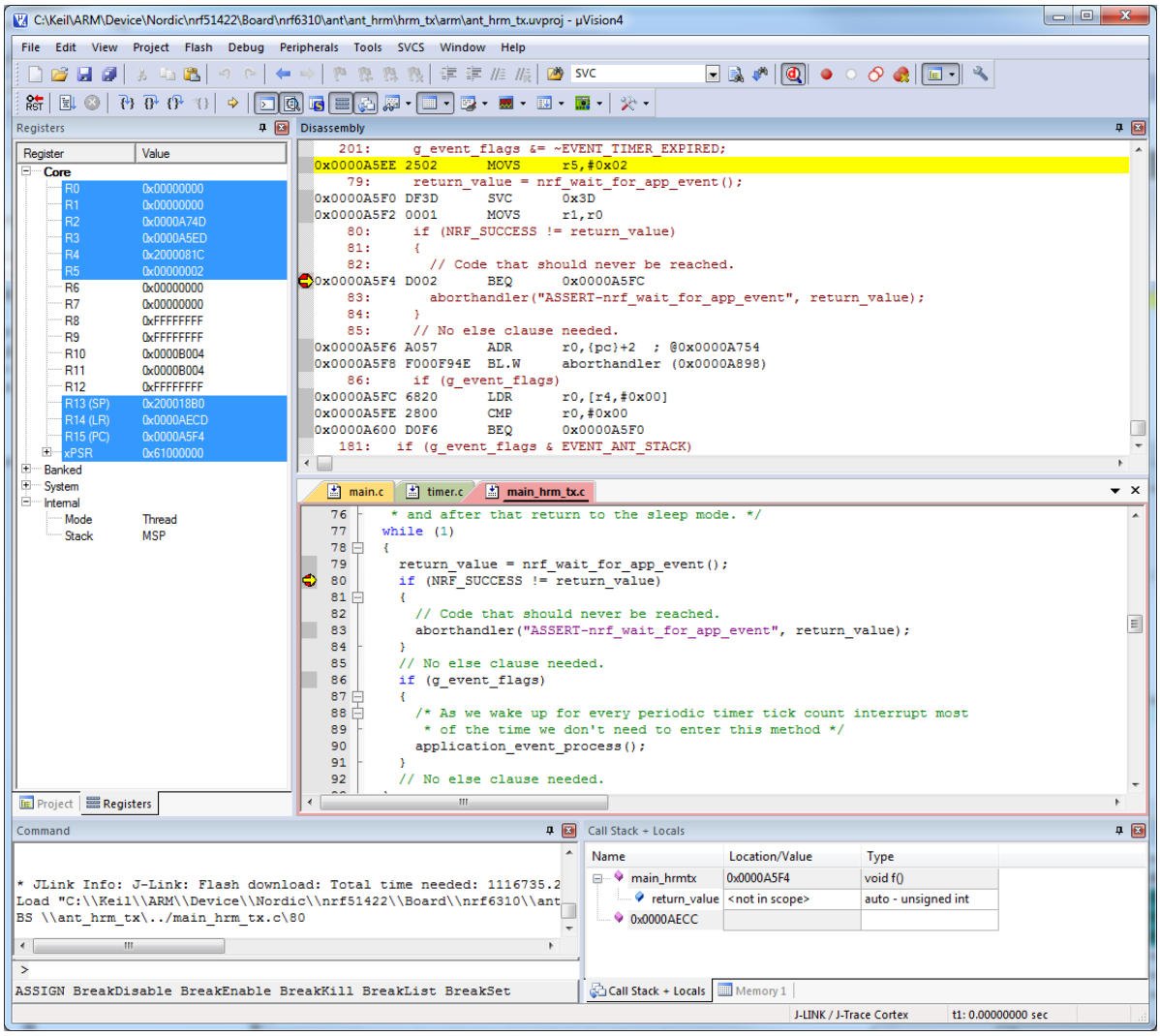


Figure 32 *Setup with breakpoint after an SVC call*

8 Software Development Kit

The nRF51 Software Development Kit (SDK) enables you to develop applications for the following protocol stacks:

- ANT
- ANT+

Note: ANT+ keys are needed to use ANT+ profiles included in the SDK. Please register as an ANT+ adopter at http://thisisant.com to get the keys.

8.1 Installing the nRF51 SDK

The nRF51 SDK is a part of the DK downloadable content available from your My Page account, see ***section 2.2 on page 5***. The SDK is downloaded as a MSI file (a windows installer) and is installed by running the application. When installing the SDK you can select: Keil MDK Support, Master Control Panel, and/or Custom install.

- **Keil MDK support** - installs Keil µVision example project files, the code memory programming algorithm for the J-Link debugger, and the Nordic nRF51 series device database file for Keil.
- **Custom install** - installs a software archive to a customized location.
- **nrfjprog** - installs a command line programming interface to be used with SEGGER debuggers.

Note: The Keil MDK Support option will only be available if you already have the Keil MDK toolchain installed.

9 Troubleshooting

The nRF51422 device on the PCA10006/7 does not respond when I try to contact it. What has happened?

Verify that the jumper on connector **K1** on the nRF51422 module is set in the position Pin 2 and Pin 3.

When I connect multiple SEGGER J-Link debugger boards to my computer, µVision does not recognize them correctly.

This is a known limitation with µVision in MDK v4.53 or earlier that is fixed in later versions. Upgrade to version 4.54 or later.

The debugger doesn't work.

Please refer to ***section 4.2.1 on page 15***.

On my 32 bit Windows XP machine, I get an error message with code 2908 when reinstalling either the nRF514 or nRF518 SDK.

Installing or reinstalling either the nRF518 SDK or the nRF514 SDK *after* the nRF518 SDK has been uninstalled will cause error message code 2908 during installation. Use the Task manager (**Ctrl**+**Shift**+**Esc**) to end the task **nRF514/8 SDK Setup**. Drivers included in nRF518 will still be installed (if not already installed).

The debugger seems to freeze while debugging.

If running a SoftDevice that has been programmed with the "Lock SoftDevice from Read back" enabled (see section ***section 6.1 on page 26***, the debugger will halt while stepping to an SVC instruction. You should set the breakpoint after the SVC instruction and run the application to the breakpoint, or step over any SVC instructions. See ***section 7.1.2 on page 40*** for more details.

> **Note:** The nRF51422 chip is pre-programmed with the S210 nRF51422 SoftDevice with **Lock SoftDevice from Read back** enabled.

Software gets out of sync while debugging.

Setting/modifying breakpoints on a running system using the SEGGER debugger will halt the CPU, which may result in software that is out of sync. You should avoid setting breakpoints while the system is running.

The debugger does not halt on breakpoints.

Some Keil projects in the SDK have **Optimization level 3 (-03)** and **Optimize for time** checked. If you are debugging an application with these settings, your breakpoint set might have no effect.

1. Press **Alt+F7** to open the Target options dialog.
2. Select **C/C++**.
3. Select **Optimization level 0** from the scroll down list.
4. Uncheck **Optimize for time**.

The debugger is not able to detect my nRF51 device after I have downloaded my firmware.

If the nRF51 device goes to `SystemOff` too soon after reset, it will have a problem communicating with the J-Link debugger. You can recover using the **Recover** button in nRFgo Studio.

1. Cycle the power to the nRF51 chip before you start the Recover application.

Note: The Recover function will erase all application firmware apart from the one that comes pre-programmed on the chip.

Appendix A: Installing drivers and configuring KEIL projects for the SEGGER debugger

This appendix describes the steps for installing the software and using the SEGGER J-Link Lite debugger with Keil µVision for nRF51 series devices, based on J-Link software version 4.52b or later.

Prerequisite

You need Keil µVision with ARM-MDK that you have tested to be working with MDK version 4.54.

Note: All projects in the nRF514 SDK are preset to work with the SEGGER debugger. Only the following step ***Download and install SEGGER drivers*** is needed.

Download and install SEGGER drivers

1. Download the latest SEGGER J-Link software and documentation pack from http://www.segger.com/jlink-software.html.
2. Download and run the J-Link Software and documentation pack for Windows from http://www.segger.com/jlink-software.html. The serial number from your SEGGER J-Link hardware is needed, see ***Figure 34 on page 46***.
3. During installation you will be prompted to select the IDE that you want updated with the latest SEGGER DLLs. Check the box for **Keil MDK** and any other IDEs you want to use with SEGGER.

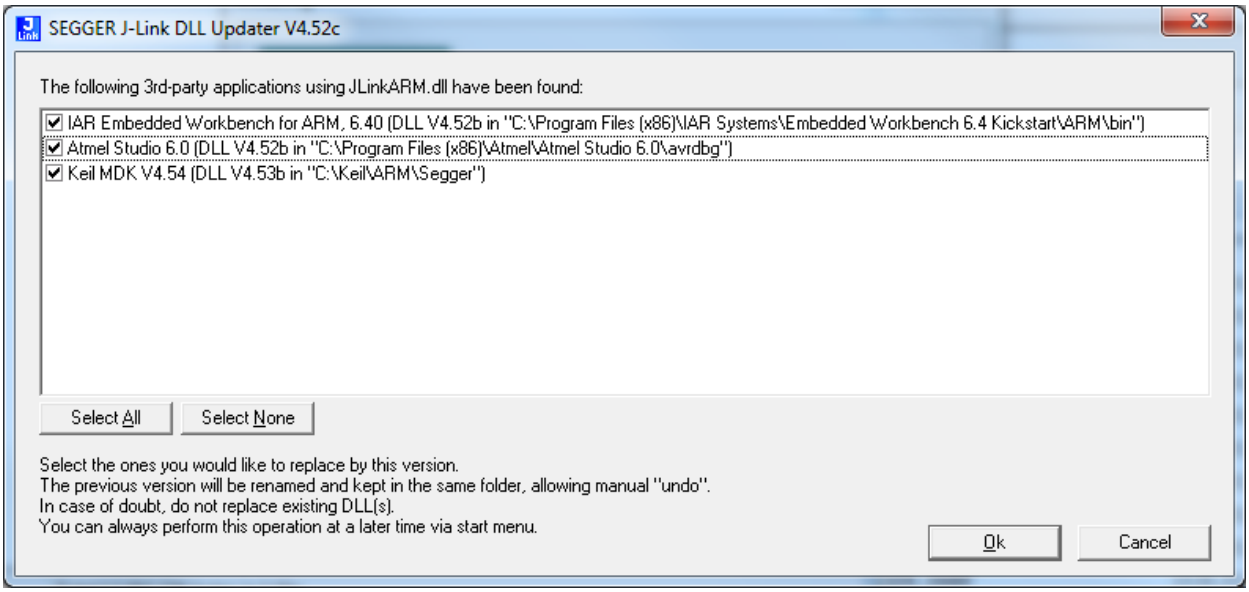

SEGGER J-Link DLL Updater V4.52c
The following 3rd-party applications using JLinkARM.dll have been found:
IAR Embedded Workbench for ARM, 6.40 (DLL V4.52b in "C:\Program Files (x86)\IAR Systems\Embedded Workbench 6.4 Kickstart\ARM\bin")
Atmel Studio 6.0 (DLL V4.52b in "C:\Program Files (x86)\Atmel\Atmel Studio 6.0\avrdbg")
Keil MDK V4.54 (DLL V4.53b in "C:\Keil\ARM\Segger")
Select All
Select None
Select the ones you would like to replace by this version.
The previous version will be renamed and kept in the same folder, allowing manual "undo".
In case of doubt, do not replace existing DLL(s).
You can always perform this operation at a later time via start menu.
Ok
Cancel

Figure 33 *IDEs selected for updating to the latest SEGGER DLLs*

4. Go to http://www.segger.com/IDE_Integration_Keil.html#knownproblems for MDK v4.54. Download JL2CM3 and copy it to **<keil>/ARM/Segger**. This patch is necessary for the SEGGER debugger to work.
5. Plug in the J-Link Lite CortexM-9 module with USB cable. The LED will blink while the driver installation occurs. Wait until the LED is continuously lit

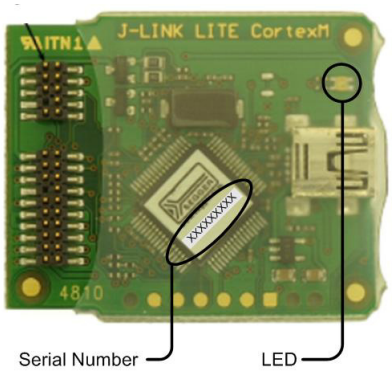


Figure 34 *J-Link Lite CortexM-9 serial number location*

Configuring KEIL projects for SEGGER debugger for first time use

Create **JLinkSettings.ini** file with the contents shown in ***Figure 40 on page 50***. The file **JLinkSettings.ini** should be saved in the same folder as Keil µVision project (uvproj) file.

1. Double-click an example project file to open the Keil µVision IDE.
2. Click **Target Options** on the toolbar or click **Project** menu and select **Options for Target**

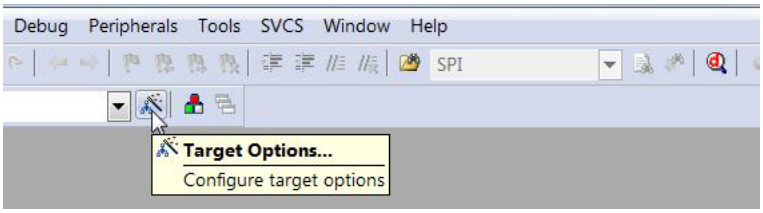


Figure 35 *Keil Target configuration*

3. Under the **Debug** tab in the Use list, select **J-LINK / J-Trace Cortex** option as shown in ***Figure 36***.
4. Click **Settings** as shown in ***Figure 36.*** Both the SEGGER Control Panel and the Keil Target Driver Setup will open.

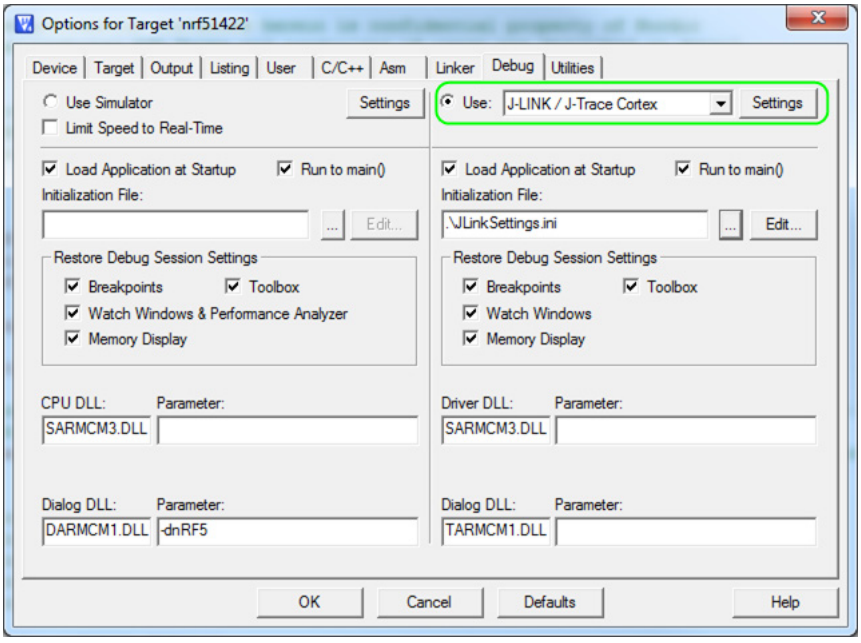


Figure 36 *Selecting J-Link debugger in Keil*

Note: If the SEGGER J-Link Lite firmware requires an update you will be prompted with the message "A new firmware version is available for the connected emulator". In this case, click **OK**.

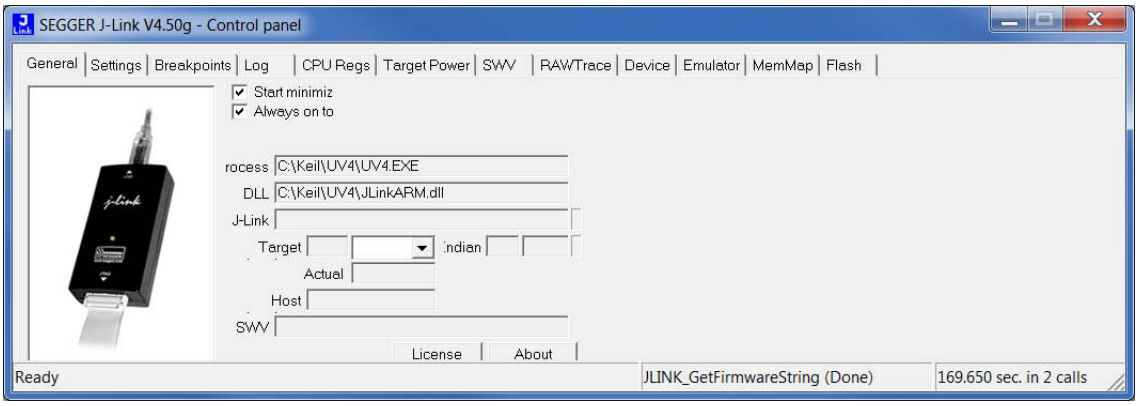


Figure 37 *SEGGER control panel*

5. Click the **Debug** tab shown in the figure. Set Port to **SW** and Max Clock to **1 MHz**, as shown in ***Figure 38***. Make sure that **SN** and **IDCODE** are populated properly and click **OK**.

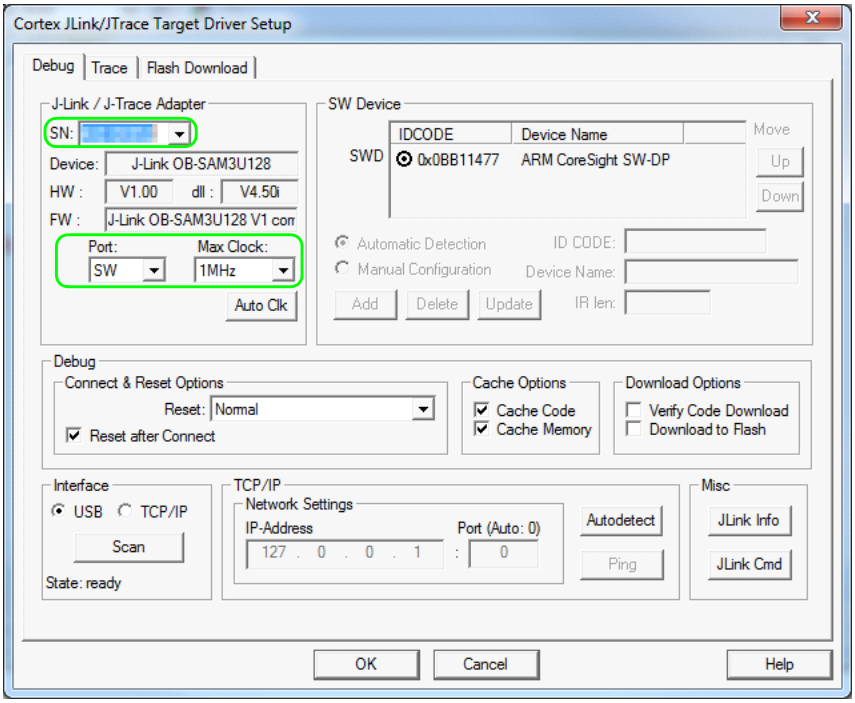


Figure 38 *Debug settings*

6. Select the J-Link device for target programing.

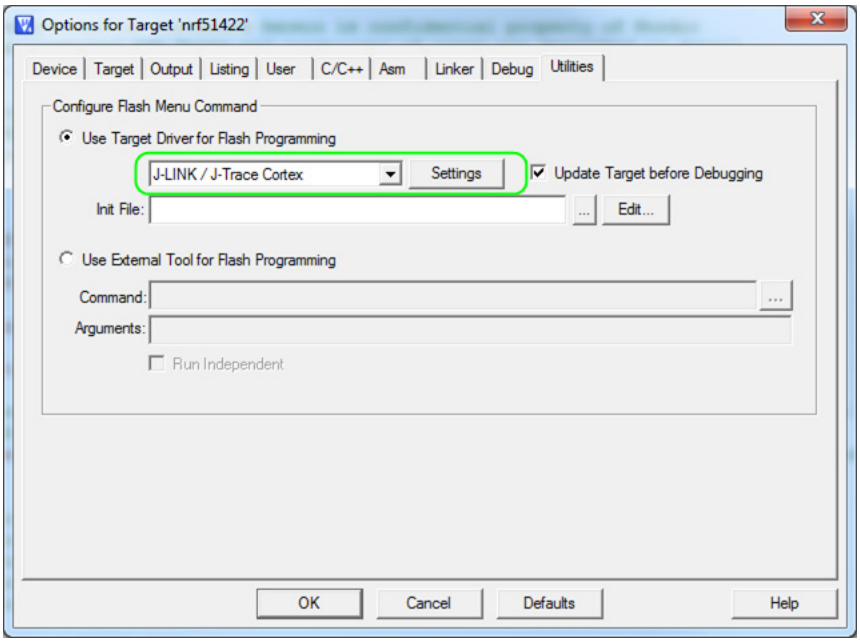


Figure 39 *Flash settings*

7. If the J-Link serial number appears in the SN field, the device is properly installed. The default settings can be accepted by clicking **OK**, closing both the SEGGER Control Panel and Keil target Driver Setup.

JLinkSettings.ini file

While debugging with a SoftDevice, change AllowSimulation = 1 to AllowSimulation = 0 in your default **JLinkSettings.ini** file under your project.

1. Enter the utilities settings from KEIL target options (**ALT**+**F7**) and click the **...** button to select the **JLinksettings.ini** file.
2. Click **Edit** and change the AllowSimulation as specified above.

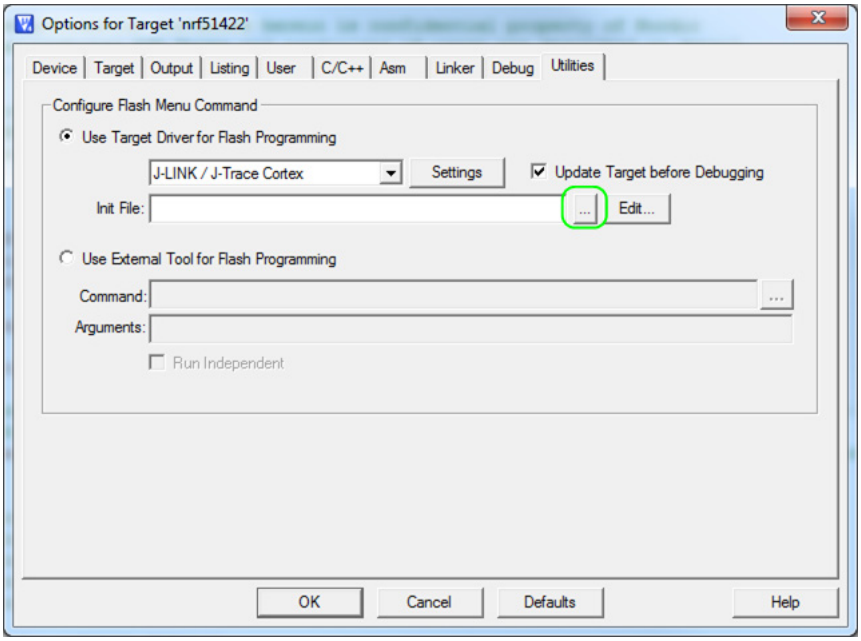


Figure 40 *Locating JLinkSettings.ini*

Liability disclaimer

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function or design. Nordic Semiconductor ASA does not assume any liability arising out of the application or use of any product or circuits described herein.

Life support applications

Nordic Semiconductor's products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

Contact details

For your nearest dealer, please see http://www.nordicsemi.com.
Information regarding product updates, downloads, and technical support can be accessed through your My Page account on our homepage.

Main office: Otto Nielsens veg 12
7052 Trondheim
Norway

Phone: +47 72 89 89 00
Fax: +47 72 89 89 89

Mailing address: Nordic Semiconductor
P.O. Box 2336
7004 Trondheim
Norway

Revision history

Date	Version	Description
February 2013	1.1	Updated content to match v2.0 of the hardware. Updated the hardware content in ***chapter 5 on page 17*** and SDK information in ***chapter 8 on page 42.***
November 2012	1.0	First release

ARM statement

Keil, µVision, and Cortex are trademarks of ARM Limited. All other brands or product names are the property of their respective holders.